

# Universidad de Alcalá

## Escuela Politécnica Superior

**Máster Universitario en Ingeniería Electrónica**

### **Trabajo Fin de Máster**

Implementación de un sistema de control y gestión de una  
microrred

**Autor:** Miguel Tradacete Ágreda

**Tutor:** Francisco Javier Rodríguez Sánchez

2020





# UNIVERSIDAD DE ALCALÁ

## ESCUELA POLITÉCNICA SUPERIOR

**Máster Universitario en Ingeniería Electrónica**

**Trabajo Fin de Máster**

**Implementación de un sistema de control y gestión de una  
microrred**

Autor: Miguel Tradacete Ágreda

Tutor: Francisco Javier Rodríguez Sánchez

**Tribunal:**

**Presidente:** Francisco Javier Meca Meca

**Vocal 1º:** Santiago Cóbreces Álvarez

**Vocal 2º:** Álvaro Hernández Alonso

Fecha de depósito: 19 de octubre de 2020



# Agradecimientos

*No nos atrevemos a muchas cosas porque son difíciles,  
pero son difíciles porque no nos atrevemos a hacerlas.*

Lucio Anneo Séneca

Este Trabajo Fin de Máster pone término a una gran etapa de estudios académicos que, además de aportarme amplios conocimientos en el ámbito de la ingeniería electrónica y automática industrial, me ha enseñado cómo seguir aprendiendo en el extenso mundo de la ingeniería, ya que esto ha sido tan solo la antesala. Este trabajo pues, delimita el comienzo de una nueva etapa donde el conocimiento crecerá gracias a la unión de la pasión y la profesión.

Quiero agradecer en primer lugar a toda mi familia por el apoyo incondicional que siempre me han dado y que sin la energía de los cuales nunca sería quien soy, y no habría llegado por tanto hasta aquí. Agradecer también a todas aquellas personas especiales que me han acompañado a lo largo de esta etapa y me han impulsado cuando los ánimos enflaquecían.

También agradecer a todos los profesores y compañeros del grupo de investigación toda la ayuda y guía que me han brindado y sin la cuál este trabajo habría sido otro muy distinto, mucho más difícil. En especial agradecer a Francisco Javier, Enrique, Toño, Miguel, Guillermo y Carlos; grandes personas y grandes profesionales con los que seguirá siendo siempre un placer y un honor trabajar a su lado.



# Resumen

En las últimas décadas las microrredes están prosperando favorecidas por la penetración de las energías renovables en el ámbito particular. Las microrredes en DC tienen ciertas ventajas sobre las microrredes AC en la integración de DER en DC, por lo que nuevas configuraciones están siendo estudiadas. El proyecto Helios Sharing de la Universidad de Alcalá (UAH) busca convertir BTSs en DERs basadas en fuentes renovables, convirtiéndolas así en microrredes en DC. A razón de este proyecto se comenzó a construir un prototipo de microrred en la Escuela Politécnica de la UAH, con una instalación fotovoltaica de 3kW. Este Trabajo de Fin de Máster forma parte del proyecto Helios Sharing y presenta la implementación hardware y software del controlador de la microrred, implementando con una Raspberry Pi 4 un sistema de gestión de energía (EMS) y un Servidor del estándar IEC61850, concretamente el IEC61850-7-420 dedicado a la comunicación en microrredes. Además se desarrolla una estación meteorológica conectada a una nube IoT y un HMI para la visualización de todos los parámetros de la microrred y su control manual. El conjunto de la microrred y el sistema de control es validado a través de pruebas reales durante días consecutivos y en distintas condiciones ambientales.

**Palabras clave:** Microrredes DC, Sistema de gestión de energía (EMS), IEC61850, Internet of Things (IoT), Diseño electrónico.



# Abstract

In the last few decades microgrids are prospering due to penetration of renewable energies in the home sector. DC microgrids have some benefits over AC microgrids integrating DC DERs, therefore new configurations are under study. The Helios Sharing project of the University of Alcalá (UAH) aims to convert BTSs into DERs based on renewable resources, converting them into DC microgrids. Consequently, the construction of a microgrid prototype started at the Polytechnic School of the UAH, with a 3kW photovoltaic system. The present Master's Thesis is part of the Helios Sharing project and presents the hardware and software implementation of the microgrid controller, implementing (with a Raspberry Pi 4) an energy management system (EMS) and a IEC61850 Server, specifically the IEC61850-7-420 which focuses on microgrid communications. In addition, this Thesis develops a weather station connected to an IoT cloud, and a HMI to display all microgrid's parameters and to manually control it. The whole microgrid and control system is validated through real tests along consecutive days and with different environmental conditions.

**Keywords:** DC microgrids, Energy management system (EMS), IEC61850, Internet of Things (IoT), Electronic design.





# Índice general

<b>Resumen</b>	<b>VII</b>
<b>Abstract</b>	<b>IX</b>
<b>Índice general</b>	<b>XI</b>
<b>Índice de figuras</b>	<b>XIV</b>
<b>Índice de tablas</b>	<b>XVII</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Contexto del trabajo . . . . .	2
1.2. Objetivos del trabajo . . . . .	4
1.3. Organización de la memoria . . . . .	4
<b>2. Revisión bibliográfica</b>	<b>5</b>
2.1. Arquitectura hardware de microrredes en DC . . . . .	5
2.2. Arquitectura software de microrredes . . . . .	9
2.2.1. Integración del estándar IEC61850 . . . . .	9
<b>3. Desarrollo</b>	<b>13</b>
3.1. Introducción . . . . .	13
3.2. Diseño electrónico . . . . .	16
3.2.1. Driver Hardware . . . . .	16
3.2.1.1. Estructura . . . . .	17
3.2.1.2. Componentes . . . . .	19
3.2.1.3. Comunicación I2C . . . . .	22
3.2.1.4. Ruido electromagnético . . . . .	23
3.2.1.5. Alimentación . . . . .	27
3.2.2. Estación meteorológica . . . . .	29
3.2.2.1. Hardware . . . . .	30
3.2.2.2. Software . . . . .	34
3.3. Diseño software . . . . .	36
3.3.1. Controlador DER . . . . .	36
3.3.1.1. Nivel 1 - Interacción física . . . . .	37
3.3.1.2. Nivel 2 - Lógica . . . . .	41
3.3.2. HMI (Nivel 3) . . . . .	45
3.3.2.1. Modo operativo . . . . .	46
3.3.2.2. Modo normal . . . . .	47

<b>4. Resultados</b>	<b>49</b>
4.1. Hardware . . . . .	49
4.2. Casos de uso . . . . .	53
4.2.1. Día soleado - 8 de septiembre de 2020 . . . . .	53
4.2.2. Día nublado - 14 de septiembre de 2020 . . . . .	56
4.3. Pruebas modo operario . . . . .	60
4.4. Información meteorológica . . . . .	62
4.4.1. Recogida en Controlador DER . . . . .	62
4.4.2. Recogida en ThingSpeak . . . . .	62
<b>5. Conclusiones y líneas futuras</b>	<b>65</b>
5.1. Conclusiones . . . . .	65
5.2. Líneas futuras . . . . .	66
<b>6. Presupuesto</b>	<b>67</b>
6.1. Coste del diseño hardware y software . . . . .	69
6.2. Coste del personal . . . . .	70
6.3. Honorarios Facultativos . . . . .	70
6.4. Presupuesto total . . . . .	70
<b>Bibliografía</b>	<b>71</b>
<b>A. Esquemáticos del Driver Hardware</b>	<b>75</b>
<b>B. Esquemáticos de la Estación Meteorológica</b>	<b>79</b>
<b>C. Modelo IEC61850 de la planta de la microrred</b>	<b>83</b>



# Índice de figuras

1.1. Relaciones en una microgrid comunitaria basada en DERs [1] . . . . .	1
1.2. Esquema conceptual de la instalación del proyecto Helios Sharing. . . . .	3
2.1. Esquema representativo del IoT [17]. . . . .	5
2.2. Especificaciones generales de la Raspberry Pi 4 modelo B [19]. . . . .	6
2.3. Dashboard Node-RED con los indicadores de consumo [2]. . . . .	7
2.4. Configuración presentada en [3]. . . . .	8
2.5. Arquitectura de la plataforma de monitorización [4]. . . . .	8
2.6. Estructura general del sistema [5]. . . . .	9
2.7. Jerarquía de un Dispositivo Electrónico Inteligente (IED) basado en el estándar IEC61850 [6]. . . . .	10
2.8. Esquema del sistema de generación solar fotovoltaica implementado en [7]. . . . .	11
2.9. Modelo del estándar IEC61850 de un grupo electrógeno, PV, BESS y cargas [8]. . . . .	12
3.1. Esquema conceptual instalación. . . . .	13
3.2. Conexiones del regulador DC-DC [13]. . . . .	14
3.3. Interrelación conceptual entre sistemas IEC61850 y periféricos. . . . .	15
3.4. Esquema conceptual del TFM. . . . .	15
3.5. Esquema conceptual del driver hardware. . . . .	16
3.6. Driver hardware: placa principal. . . . .	18
3.7. Driver hardware: SBC, ADC y BME680. . . . .	19
3.8. Conexionado de los relés de enclavamiento. . . . .	19
3.9. Ruido electromagnético: tipos de acoplamiento [39]. . . . .	23
3.10. Disposición inicial del driver hardware. . . . .	24
3.11. Estudio del ruido conducido por la masa del sistema. . . . .	26
3.12. Estudio del ruido conducido por el divisor resistivo ( $R_m/R_A$ ). . . . .	26
3.13. Área de bucle con máxima interferencia ( $\cos\theta = 90$ ) [38]. . . . .	27
3.14. Cables trenzados. Transmisión de datos y alimentación. . . . .	28
3.15. Disposición de las masas del driver hardware. . . . .	28
3.16. Driver hardware: versión final. . . . .	29
3.17. Estación meteorológica. . . . .	30
3.18. Logotipo MQTT [43]. . . . .	30
3.19. Distribución conceptual de los componentes de la estación meteorológica. . . . .	31
3.20. Estación meteorológica. . . . .	32
3.21. Acondicionamiento del anemómetro y veleta. . . . .	32
3.22. Montaje de la RTD en el módulo fotovoltaico. . . . .	33
3.23. Electrónica de adquisición: montaje. . . . .	34
3.24. Logotipo FreeRTOS [48]. . . . .	34

3.25. Esquema conceptual del programa de la estación meteorológica. . . . .	35
3.26. Esquema conceptual del Controlador DER. . . . .	36
3.27. Logotipo ROS [51]. . . . .	37
3.28. Esquema conceptual del nodo ROS lectura de sensores. . . . .	38
3.29. Estimación del SoC a partir de la tensión en circuito abierto. . . . .	39
3.30. Conexión Modbus: Raspberry Pi 4. . . . .	40
3.31. Puerto de comunicación RS232 - J12 (6 pines) [13]. . . . .	41
3.32. Esquema conceptual del nodo ROS Servidor IEC61850. . . . .	43
3.33. Transiciones de la máquina de estados (FSM) del EMS. . . . .	45
3.34. Logotipo de Node-RED. . . . .	46
3.35. Programación por bloques en Node-RED. . . . .	46
3.36. Dashboard principal de Node-RED. . . . .	48
4.1. Driver hardware - Placa principal. . . . .	50
4.2. Elementos en el panel metálico no pertenecientes a la placa principal. . . . .	50
4.3. Conjunto de la instalación (interior de la caseta). . . . .	51
4.4. Instalación parte superior. . . . .	52
4.5. Estación meteorológica - Electrónica de adquisición. . . . .	52
4.6. Potencia fotovoltaica vs Irradiancia. . . . .	53
4.7. Parámetros de las baterías (BESS). . . . .	54
4.8. Corrientes de la microrred. . . . .	55
4.9. Estados y variables de control del EMS. . . . .	55
4.10. Parámetros de los módulos fotovoltaicos. . . . .	56
4.11. Potencia fotovoltaica vs Irradiancia. . . . .	57
4.12. Parámetros de las baterías (BESS). . . . .	58
4.13. Corrientes de la microrred. . . . .	58
4.14. Estados y variables de control del EMS. . . . .	59
4.15. Parámetros de los módulos fotovoltaicos. . . . .	60
4.16. Modo operario: Secuencia de finalización de pruebas. . . . .	61
4.17. HMI en modo operario. . . . .	61
4.18. Información meteorológica - SBC (Raspberry Pi 4). . . . .	62
4.19. Información meteorológica - Nube IoT (ThingSpeak). . . . .	63
4.20. Irradiancia - Nube IoT (ThingSpeak). . . . .	63
4.21. Velocidad y dirección del viento - Nube IoT (ThingSpeak). . . . .	63



# Índice de tablas

3.1. Expansor I2C - Asignación del registro 8 bit. . . . .	22
3.2. Estados de la máquina de estados (FSM) del EMS. . . . .	44
6.1. Coste de los materiales para el diseño hardware y software. . . . .	69
6.2. Coste del personal. . . . .	70
6.3. Coste de honorarios facultativos. . . . .	70
6.4. Coste total del trabajo. . . . .	70





# Índice de ecuaciones

3.6. Densidad de flujo magnético . . . . .	24
3.7. Tiempo de pico . . . . .	25
3.8. Frecuencia natural . . . . .	25
3.9. Flujo magnético . . . . .	25
3.10. Tensión de ruido (radiada y de subida) . . . . .	25
3.11. Tiempo de pico (bajada) . . . . .	25
3.12. Frecuencia natural (bajada) . . . . .	25
3.13. Tensión de ruido (radiada y de bajada) . . . . .	25
3.14. Tensión de ruido (conducida) . . . . .	26
3.15. Consumo Raspberry Pi 4 . . . . .	29
3.16. Consumo placa principal . . . . .	29
3.17. Consumo estación meteorológica . . . . .	34
3.18. Contador de Coulombios . . . . .	38



# Lista de acrónimos

AC	Alternating Current.
BESS	Battery Energy Storage System.
BTS	Base Transceiver Station.
CO2	Dióxido de carbono.
DC	Direct Current.
DER	Distributed Energy Resources.
DO	Data Object.
EMC	Electromagnetic Compatibility.
EMS	Energy Management System.
ESS	Energy Storage System.
FSM	Finite State Machine.
GND	System ground.
GOOSE	Generic Object Oriented Substation Events.
GPIO	General-purpose Input/Output.
HMI	Human-Machine Interfaces.
I2C	Inter-Integrated Circuit.
IED	Intelligent Electronic Device.
IoT	Internet of Things.
LD	Logical Device.
MMS	Manufacturing Message Specification.
MQTT	Message Queuing Telemetry Transport.
NC	Normally Closed.
NL	Nodo Lógico.
PLC	Programmable Logic Controller.
ROS	Robotic Operating System.
RTD	Resistance Temperature Detector.
RTU	Remote Terminal Unit.
SBC	Single Board Computer.

SoC	State of Charge.
TCP/IP	Transmission Control Protocol/Internet Protocol.
TFM	Trabajo Fin de Máster.
USB	Universal Serial Bus.
VCC	Voltage at the Common Collector.
VPP	Virtual Power Plant.





# Introducción

Desde el inicio de las redes eléctricas la arquitectura de distribución de energía se ha mantenido sin modificar hasta la actualidad, produciéndose la energía en grandes centrales eléctricas y siendo enviada a los consumidores de forma unidireccional [9]. En los últimos años la generación y distribución de energía está evolucionando rápidamente, impulsada en gran parte por la aparición de las energías renovables. Aunque estas energías no son nuevas, ahora son accesibles y económicas para consumidores particulares y grandes empresas. De este modo, cada vez más personas optan por instalarlas en viviendas, edificios o en diferentes emplazamientos alejados de núcleos urbanos. Estas instalaciones en su mayoría son dependientes de la red eléctrica (pudiendo también funcionar de forma aislada) y pueden llegar a producir un excedente de energía que es inyectado a la red eléctrica. Aparecen así las microrredes, basadas unidades de generación distribuida (DER) [10]. En la figura 1.1 se muestra el esquema básico de una microrred comunitaria con conexión a la red eléctrica, destacando la aparición del término *prosumer* referida a aquellos consumidores de energía que también tienen capacidad de producirla.

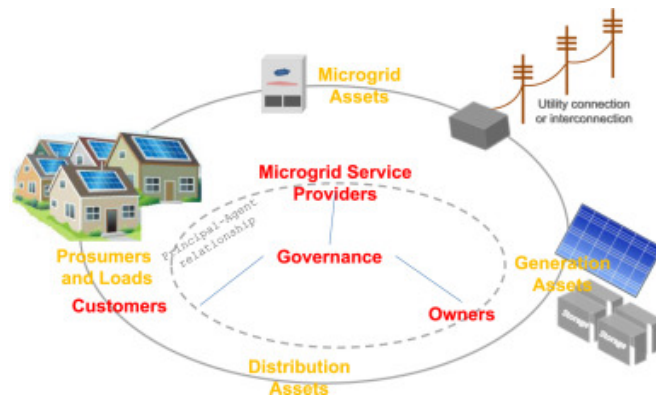


Figura 1.1: Relaciones en una microgrid comunitaria basada en DERs [1]

Las microrredes se consolidan de este modo como una alternativa al sistema actual de distribución energética, pudiendo abastecer de energía regiones geográficas con o sin acceso a una red eléctrica principal. Pero la generación a partir de fuentes de energía renovable presenta una gran dependencia de factores no controlables como la meteorología o el tiempo. Es por ello que las microrredes necesitan sistemas de almacenamiento (ESS) y de gestión de la energía y los recursos (EMS) [11]. Para dicha gestión debe estar presente un sistema de control que gestione la carga y descarga del sistema de almacenamiento, que estime el perfil de generación eléctrica y que controle las cargas asociadas a la microrred.

## 1.1. Contexto del trabajo

Este trabajo forma parte del proyecto Helios Sharing de la Universidad de Alcalá (UAH). Dicho proyecto consiste en la reconversión de las estaciones base de comunicaciones (BTS) para que funcionen como una DER con generación eléctrica procedente de fuentes renovables, almacenamiento con baterías (BESS), conexión a la red eléctrica y cargas gestionables. Esta reconversión es aplicada a instalaciones ya existentes, aportando conexión bidireccional a la red eléctrica, generación fotovoltaica, sistemas de medida inteligente y el software de gestión necesario.

De esta manera se alcanzan dos objetivos: disminuir el coste energético de las BTS (reduciendo así también las emisiones de CO<sub>2</sub>) y proporcionar a la red eléctrica su capacidad de almacenamiento. Además en el proyecto Helios Sharing se propone que todas las unidades DER creadas a partir de las BTS dispuestas por todo el territorio español actúen como una única planta de generación (o respaldo) de grandes dimensiones. A este concepto se le denomina Virtual Power Plant (VPP).

A razón del proyecto Helios Sharing, se ha realizado una instalación fotovoltaica con la finalidad de realizar un prototipo completo de una BTS. Dicha instalación se ubica en el ala Oeste de la Escuela Politécnica de la UAH. Consta de 9 módulos fotovoltaicos [12] (una matriz de 3 módulos en serie y 3 en paralelo), un regulador DC-DC [13] que acondiciona la energía obtenida de los módulos, un grupo de 4 baterías de plomo-ácido [14], cargas de continua y un equipo de fuerza [15] actuando como rectificador unidireccional para alimentar las cargas desde la red AC (no realizará vertido de energía a la red eléctrica). En la figura 1.2 se muestra el esquema conceptual de dicha instalación, donde el rectificador unidireccional se encontraría dentro del equipo de fuerza.

La energía solar fotovoltaica producida se lleva a una caseta donde se encuentran el resto de componentes. Mediante estos elementos se alimentan unas cargas de potencia constante, simulando las antenas y distintas cargas de una BTS. La caseta se encuentra cerca de los módulos fotovoltaicos, evitando caídas de voltaje y pérdida de potencia.

El TFM que se presenta en esta memoria forma parte del proyecto Helios, y cuyo objetivo final es proporcionar controlabilidad al conjunto de la instalación y convertirla así en una microrred. Además incluye la interconexión de los diferentes elementos de la instalación, ya que hasta el momento de la realización de este trabajo solamente se habían realizado pruebas básicas. De este modo la BTS estaría capacitada para desconectarse de la red AC a conveniencia (si se está produciendo suficiente energía fotovoltaica), conectarse a ella si se requiriera de una mayor potencia o el precio del kWh de la red bajase, o se podría guardar excedente de la producción fotovoltaica para usarlo durante la noche. Estos son algunos ejemplos de las funcionalidades que este TFM aporta al prototipo del proyecto Helios Sharing.



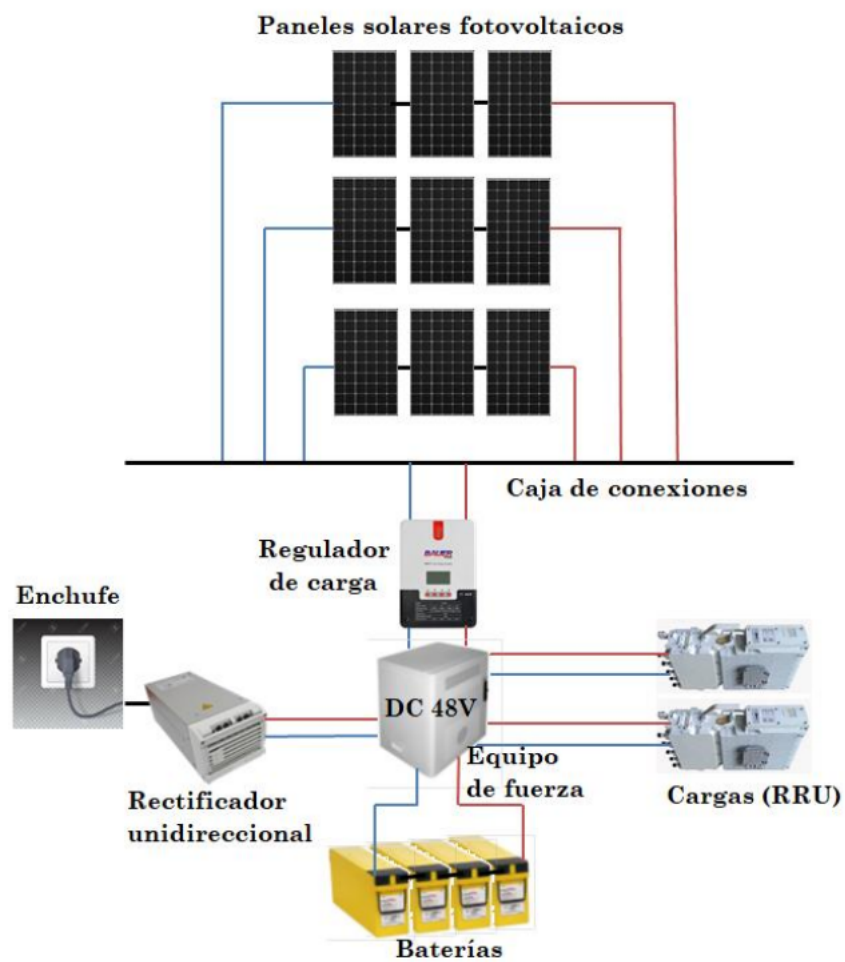


Figura 1.2: Esquema conceptual de la instalación del proyecto Helios Sharing.

## 1.2. Objetivos del trabajo

El objetivo principal de este trabajo es controlar el flujo de energía de una BTS con apoyo de fotovoltaica, convirtiéndola así en una microrred basada en DER de fuentes renovables. Los objetivos más concretos de este trabajo se exponen a continuación:

- Concluir la instalación fotovoltaica relacionada al proyecto Helios Sharing.
- Diseñar una electrónica de adquisición de datos y control sobre los elementos de la instalación.
- Implementar un HMI (Human-Machine Interfaces) para visualizar y controlar manualmente los flujos de energía de la microrred.
- Crear un controlador DER para automatizar el funcionamiento de la microrred.
- Implementar un servidor del estándar IEC61850 dentro del controlador de la microrred.
- Construir una estación meteorológica para la recogida de datos ambientales.
- Implementación de un EMS de prueba para verificar el funcionamiento global de la microrred.

Como objetivos futuros para otros trabajos que den continuidad a este TFM, se pueden destacar: modificar la instalación para poder realizar vertido a la red eléctrica y aumentar la funcionalidad de la microrred, y realizar una gestión inteligente de las baterías para aumentar su vida útil y también ahorrar dinero cargándolas o descargándolas en función del precio del kWh de la red eléctrica.

## 1.3. Organización de la memoria

Esta memoria se organiza en seis capítulos: el primero "Introducción" donde se presenta el TFM, el segundo "Revisión bibliográfica" donde se hace un repaso del estado del arte en microrredes en DC y el estándar IEC61850, el tercero "Desarrollo" donde se expone el desarrollo teórico y práctico del TFM, el cuarto "Resultados" donde se exponen los resultados experimentales obtenidos del funcionamiento de la microrred, el quinto "Conclusiones y líneas futuras" donde se comentan impresiones finales sobre el desarrollo del TFM y posibles ampliaciones a este trabajo, y el sexto "Presupuesto" donde se realiza el desglose del coste de este trabajo. Como anexos se han incluido esquemáticos y algunos de los códigos de programación utilizados.

# Revisión bibliográfica

## 2.1. Arquitectura hardware de microrredes en DC

Dentro de las microrredes es necesario implementar la algoritmia de los EMS de forma fácil y eficiente ya que es una de las piezas más importantes en su funcionamiento. Hoy en día las microrredes se están comenzando a instalar en las *smart homes* en conjunto con otros dispositivos electrónicos, integrados todos a través del *Internet of Things* (IoT). Entonces, reducir la complejidad computacional, el consumo de recursos y el coste del procesador se convierte en un punto principal, además de hacer todo esto compatible con el paradigma del IoT [16]. En este trabajo citado se muestra cómo un EMS maneja los flujos de energía entre las cargas de una *smart home*, los generadores de energía renovable, los sistemas de almacenamiento y la red eléctrica. Todos los dispositivos se comunican a través de una red inalámbrica mediante una base de datos en la nube. En la figura 2.1 se muestra la interconexión conceptual de distintos sistemas formando el IoT, desapareciendo así las comunicaciones de un solo canal entre dispositivos.



Figura 2.1: Esquema representativo del IoT [17].

Existen múltiples opciones comerciales de procesadores adecuados para la implementación de los sistemas de control y de comunicaciones en microrredes dentro de las *smart homes*. Algunos de estos procesadores son: NVIDIA Jetson Nano con gran capacidad de computación pero con un precio por encima de los 100€, ASUS Tinker Board S con unas prestaciones más ajustadas al propósito de las

microrredes pero con un precio alrededor de los 80€, ODR0ID-XU4 con unas prestaciones adecuadas y con un precio inferior a los 50€ pero sin módulo Wi-Fi integrado (crucial para el IoT), y finalmente la Raspberry Pi 4 modelo B con unas características adecuadas incluyendo módulo Wi-Fi y Bluetooth y con un precio inferior a 50€. En este TFM se decide utilizar la Raspberry Pi 4 modelo B ya que reúne todas las características necesarias (incluido un precio relativamente bajo) y es la más utilizada en la bibliografía revisada, suponiendo mayor documentación disponible en comparación al resto de procesadores mencionados.

La Raspberry Pi es un grupo de ordenadores de placa simple (SBC) de coste reducido, fabricados por la Raspberry Pi Foundation (UK). Su propósito es proporcionar a todo el mundo la capacidad de crear proyectos digitales e informáticos [18]. La Raspberry Pi no incorpora ni teclado ni ratón ni ningún periférico, aunque se han creado muchos accesorios tanto oficiales como no oficiales.

Raspberry Pi cuenta con su propio sistema operativo de software libre. Este sistema operativo es una distribución adaptada de Debian, llamada "*Raspberry Pi OS*" (aunque anteriormente denominada "*Raspbian*"). La Raspberry Pi también puede usar otros sistemas operativos, incluso una versión adaptada de "*Windows 10*". Todas las versiones de Raspberry Pi cuentan con un procesador Broadcom, GPU, memoria RAM, puerto HDMI, puertos USB y puerto Ethernet (el primer modelo no disponía de Ethernet), 40 pines GPIO (a partir de la Raspberry Pi 2) y un conector dedicado a la conexión de cámaras. La memoria debe ser adjuntada de forma externa. En su primera versión esta memoria debía ser una tarjeta SD y en las versiones posteriores una tarjeta microSD. En la figura 2.2 se detallan algunas de estas especificaciones referenciadas en la placa de la Raspberry Pi 4.

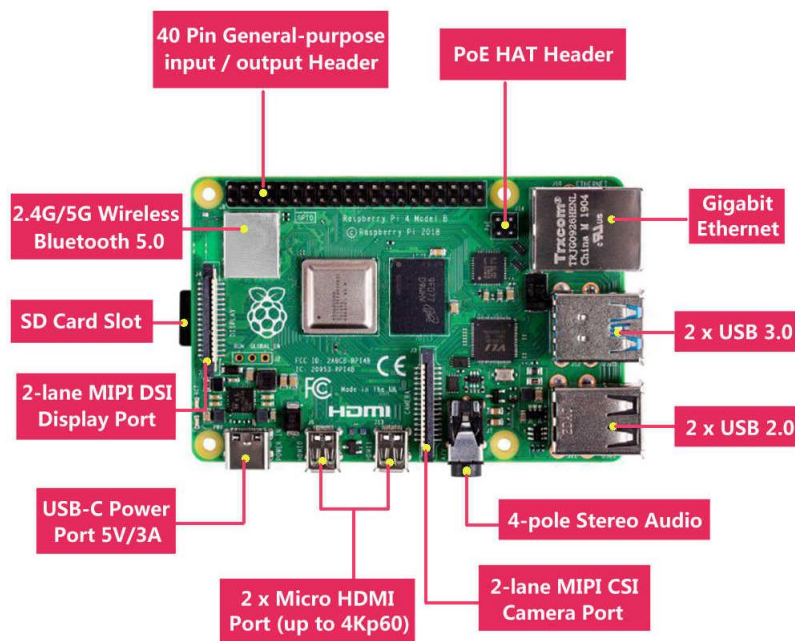


Figura 2.2: Especificaciones generales de la Raspberry Pi 4 modelo B [19].

La última versión de la Raspberry Pi es la Raspberry Pi 4 modelo B (anunciada en junio de 2019), evolución de la Raspberry Pi 3. Esta última versión es capaz de manejar dos pantallas 4K a 60 Hz. Además en esta versión se ha incluido el USB 3.0, y el puerto Ethernet no está limitado a 300 Mbps. Su nuevo procesador Broadcom es hasta tres veces más eficiente que el de la Raspberry Pi 3.

En el estado del arte actual diversos trabajos relacionados con microrredes en DC basadas en fuentes renovables y en la simulación de sus algoritmos de control, han hecho uso de la Raspberry Pi como procesador principal para llevar a cabo tareas de monitorización, comunicación, y control y optimización

de todos los elementos de las microrredes (generación, almacenamiento, conexión con la red eléctrica, etc.) [20],[21],[22].

Otros trabajos han implementado sistemas de control de energía en ámbitos reales. En [2] se presenta la optimización del consumo de energía en una escuela mediante un EMS (implementado en una Raspberry Pi) haciendo uso del IoT para recoger información de diversos sensores. El ahorro de energía se lleva a cabo controlando los aires acondicionados y la iluminación. Los sensores adquieren datos ambientales (temperatura, humedad, luminosidad, calidad del aire y movimiento). Todos estos datos son monitorizados en una interfaz gráfica mediante Node-RED (mostrada en la figura 2.3). Esta interfaz permite la identificación de patrones para mejorar el ahorro energético e incluso actuar en las cargas eléctricas de forma manual.

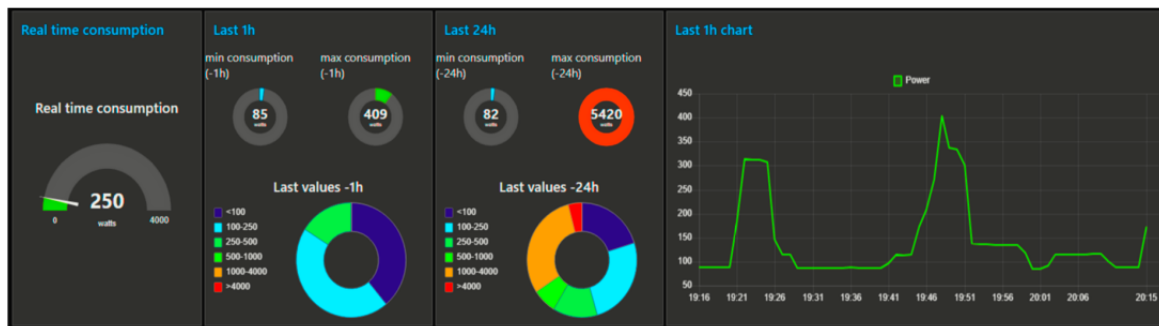


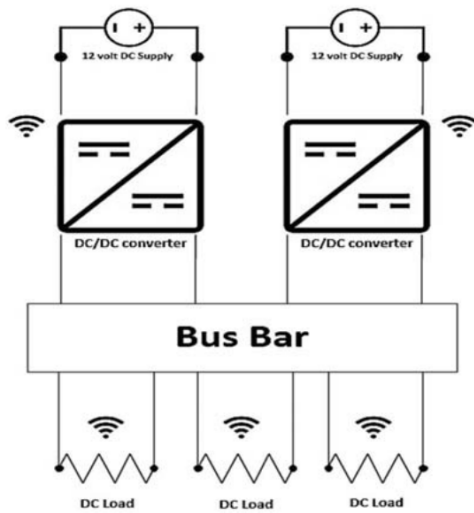
Figura 2.3: Dashboard Node-RED con los indicadores de consumo [2].

Para el procesamiento de los distintos sensores ambientales hacen uso de microcontroladores con procesador ESP32. Este procesador *dual-core* cuenta con Wi-Fi integrado y conectividad Bluetooth de largo alcance. Además cuenta con memoria interna, memoria externa y periféricos. Las dos CPUs tienen los nombres de "PRO\_CPU" y "APP\_CPU" (de "protocolo" y "aplicación"), aunque en la mayoría de aplicaciones las dos CPUs son intercambiables [23].

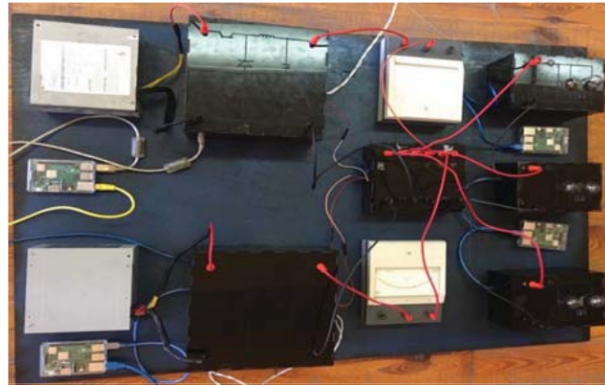
En otro trabajo [3] se expone el diseño de un prototipo experimental de microrred en DC haciendo uso de un sistema de medida basado en IoT. El sistema desarrollado (figura 2.4) simula una microrred en DC con fuentes de alimentación, convertidores DC-DC reductores, un bus DC y tres cargas con diferente categoría cada una (crítica, esencial y no prioritaria). Estas tres cargas son conectadas o desconectadas atendiendo a diferentes escenarios. El objetivo es realizar una comunicación bidireccional entre las cargas y los convertidores DC-DC para satisfacer la demanda de las cargas a la vez que se cumplen las restricciones del sistema. El procesador principal usado es una Raspberry Pi 3B y utiliza Node-RED como interfaz gráfica en la nube.

Otro artículo [4] expone una arquitectura para una microrred de ámbito residencial. Dicha microrred cuenta con generación fotovoltaica (265W), almacenamiento con baterías (25Ah) y un sistema de ventilación compuesto por dos ventiladores de 24W. Además, se instala una estación meteorológica con un anemómetro, una veleta y un sensor de temperatura, humedad y presión atmosférica. Estos sensores están conectados a un microcontrolador que recolecta los datos y los transmite al procesador central de la microrred (una Raspberry Pi) donde se guardan y visualizan. Una tarea adicional de la Raspberry Pi es procesar los datos usando una plataforma IoT/Big-data. En la figura 2.5 se muestra el esquema de la arquitectura de este artículo con imágenes de su implementación real.

En otro trabajo [5] se expone un sistema híbrido *stand-alone* formado por módulos fotovoltaicos y un aerogenerador asociados con un grupo electrógeno de reserva y almacenamiento con baterías. La estructura de la microrred AC/DC se muestra en la figura 2.6. Las baterías y los módulos fotovoltaicos están conectados a un bus DC, y el aerogenerador, el grupo electrógeno y las cargas a un bus AC. El



(a) Esquema del sistema.



(b) Montaje experimental.

Figura 2.4: Configuración presentada en [3].

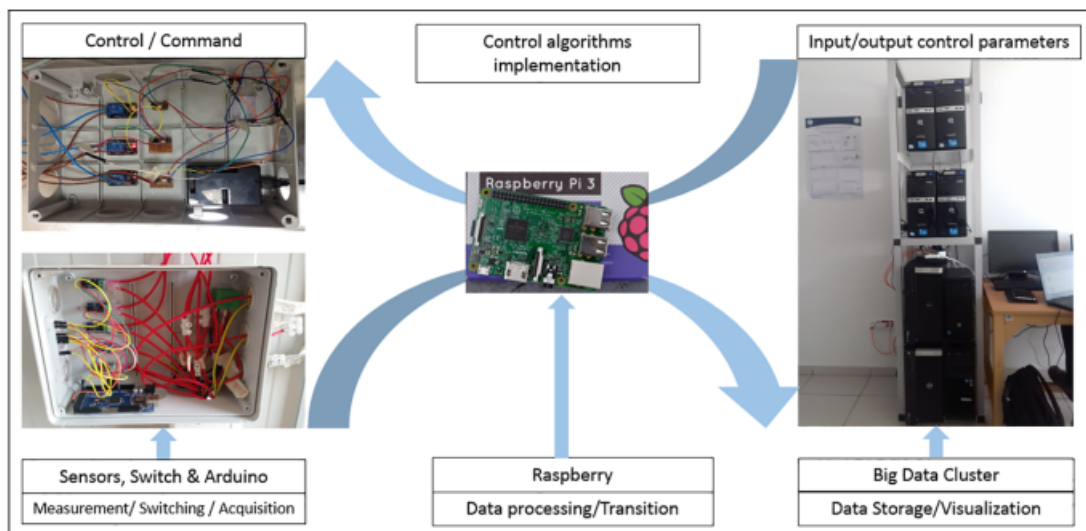


Figura 2.5: Arquitectura de la plataforma de monitorización [4].

algoritmo de gestión es simulado en LabVIEW usando datos reales de irradiancia y velocidad del viento. El EMS se implementa en una Raspberry Pi 3, a la cual se conectan las fuentes de alimentación y las cargas a través de microcontroladores Arduino. Los pines GPIO de la Raspberry Pi controlan los relés de las fuentes y las diferentes cargas.

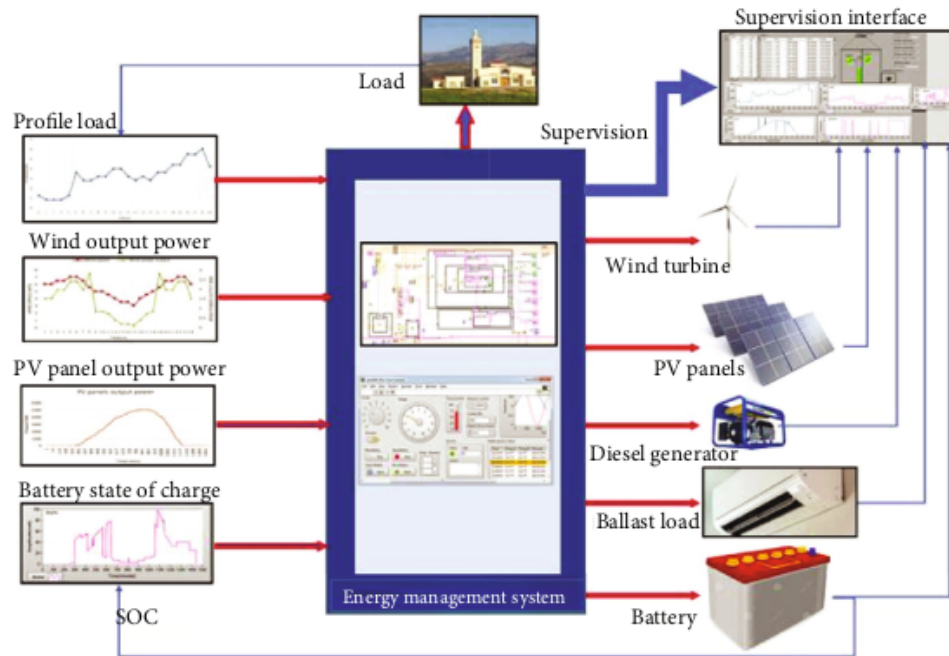


Figura 2.6: Estructura general del sistema [5].

## 2.2. Arquitectura software de microrredes

### 2.2.1. Integración del estándar IEC61850

En 2004, el estándar IEC61850 fue publicado como un estándar global para el control y la protección de los sistemas de conmutación de medio y alto voltaje. Cubre tanto los modelos de estandarización IEC como ANSI [24]. Este nuevo estándar garantiza:

- Estándares unificados para todas las subestaciones y centrales eléctricas.
- Un formato común para la descripción de las subestaciones y hacer más fácil su diseño.
- Definición de los servicios principales para la transmisión de datos usando diferentes protocolos.
- Interoperabilidad entre dispositivos de distintos fabricantes.

El estándar IEC61850 proporciona un marco de trabajo estandarizado para la integración de requisitos específicos de comunicaciones para subestaciones, características funcionales, estructura de datos en los dispositivos, unificación de los nombres de los paquetes de datos, cómo las aplicaciones interactúan con el control de los dispositivos y determina tests estandarizados [25].

En cuanto a la organización de datos se refiere, el IEC61850 presenta una metodología de modelado jerárquico, usando Nodos Lógicos (NLs) como elemento más pequeño que describe una función determinada y con la facultad de intercambiar datos entre clientes y servidores. La figura 2.7 muestra la arquitectura jerárquica funcional para el modelado de un dispositivo IED (Intelligent Electronic Device).



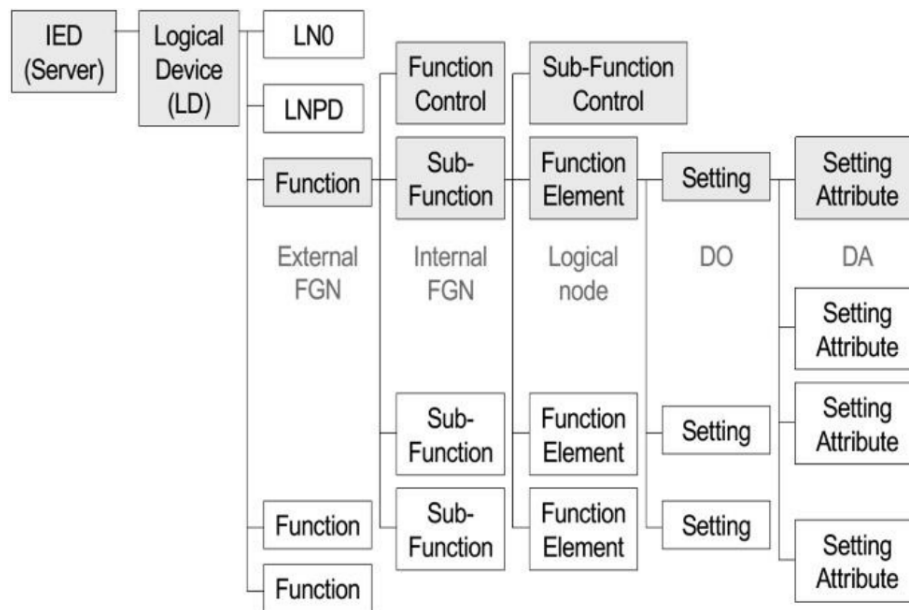


Figura 2.7: Jerarquía de un Dispositivo Electrónico Inteligente (IED) basado en el estándar IEC61850 [6].

Dentro del IEC61850 encontramos el IEC61850-7-420, dedicado a la comunicación en microrredes. Esta extensión del estándar incluye la integración de DERs en la comunicación de las estructuras de los sistemas de potencia [26]. Propone los NLs para describir completamente las DERs y los sistemas de control asociados a ellas.

La motivación principal para aplicar este estándar en el diseño de sistemas de comunicación es doble [7],[27]:

- Flexibilizar el diseño siguiendo una norma internacional aceptada.
- Buscar la interoperabilidad con el resto de dispositivos IEC61850 y los sistemas SCADA.

Algunas de sus principales características y ventajas son [28]:

- Uso de modelos virtuales.
- Uso de nombres para todos los datos.
- Todos los nombres de los objetos están estandarizados y definidos en un contexto de sistemas de potencia.
- Los dispositivos tienen la capacidad de auto-describirse.
- Servicios y funcionalidades de alto nivel.
- Bajo coste de instalación.
- Bajo coste de la migración de equipos.
- Bajo coste en la extensión de una instalación.
- Implementa nuevas capacidades.



Muchos proyectos en la actualidad hacen uso de este estándar. Un ejemplo se muestra en el trabajo [7] donde se implementan una serie de NLs de sistemas de generación basados en DER, en una SBC que adquiere y almacena distintas mediciones de una instalación fotovoltaica con baterías (en base al estándar IEC 61850-7-420). En la figura 2.8 se muestra el esquema de la instalación fotovoltaica de este trabajo. La instalación está compuesta por un conjunto de módulos fotovoltaicos, un convertidor DC-DC (con MPPT), una serie de baterías, un inversor y cargas eléctricas tanto en AC como en DC. En la figura 2.8 se muestran también los NLs implementados, interrelacionándose en función del subsistema al que pertenecen. Los subsistemas definidos son la conexión a la red, los módulos fotovoltaicos, el almacenamiento y la alimentación de las cargas eléctricas.

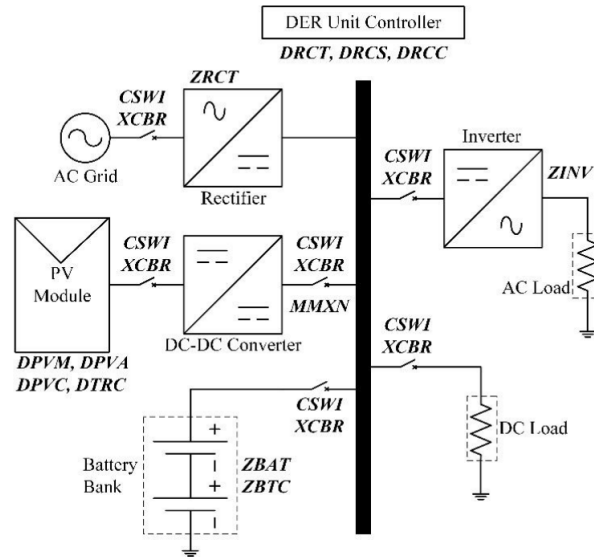


Figura 2.8: Esquema del sistema de generación solar fotovoltaica implementado en [7].

Otro ejemplo es el trabajo [8] donde se presenta la estandarización de las comunicaciones de un EMS de una microrred real. Los componentes individuales son desarrollados con Dispositivos Lógicos (LDs) y LNs. Estos modelos individuales son combinados para crear el modelo del EMS. Además, Data Objects (DOs) son utilizados para mapear todas las variables que son monitorizadas por la microrred. En cuanto a la comunicación se refiere, se utilizan mensajes GOOSE y MMS para intercambiar información entre la microrred y el EMS.

La microrred implementada cuenta con generación fotovoltaica, un grupo electrógeno, almacenamiento con baterías (BESS) y distintas cargas eléctricas. El modelo IEC61850 de este proyecto se muestra en la figura 2.9.

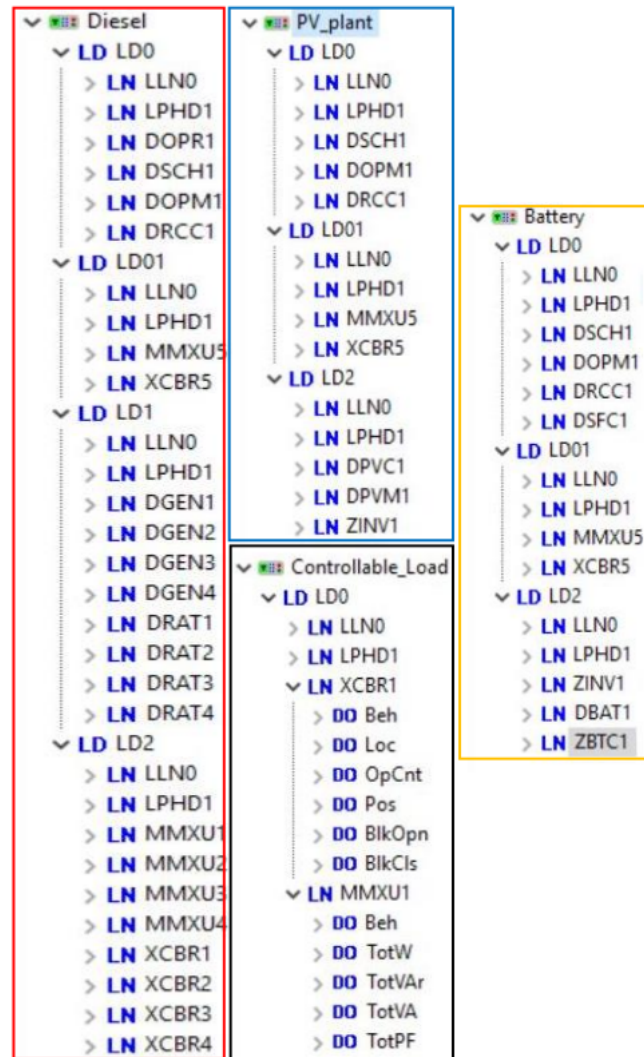


Figura 2.9: Modelo del estándar IEC61850 de un grupo electrógeno, PV, BESS y cargas [8].

# Capítulo 3

## Desarrollo

### 3.1. Introducción

#### *Descripción de la instalación*

El objetivo principal de este trabajo es controlar el flujo de energía de una BTS con apoyo de fotovoltaica, contando además con un equipo de fuerza [15] para alimentar unas antenas que consumen 6kW. En la instalación realizada como prototipo estas antenas se sustituyen por 6 cargas DC de  $3.2\Omega$  cada una (equivalente a 1kW a tensión nominal). Tres de estas cargas se establecen fijas, alimentadas desde el equipo de fuerza, y las otras tres se convierten en regulables a través de unos transistores. Se dispone también de 4 baterías conectadas en serie, formando un solo BESS.

Los módulos fotovoltaicos (descritos en la sección 1.1) se encuentran conectados al regulador DC-DC, el cual baja la tensión y se conecta al bus DC (figura 3.1) a través de su salida preparada para conectar baterías. Por otra parte, el equipo de fuerza se encuentra conectado a la red desde el lado de AC y conectado al bus DC a través de su salida dedicada a la conexión de baterías. Su conexión al bus DC se controla a través de un relé de enclavamiento. Además, cuenta con tres cargas fijas conectadas a unos conectores especialmente dedicados a este propósito e independientes eléctricamente.

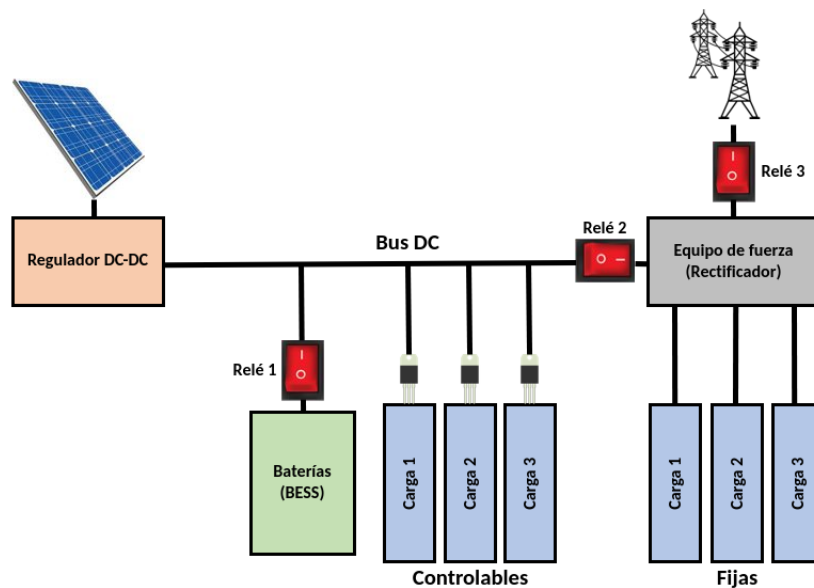


Figura 3.1: Esquema conceptual instalación.

La salida DC de baterías del equipo de fuerza está preparada para un BESS de 48V nominales y máxima tensión 57V. Por ello, el regulador DC-DC se configura para que la tensión nominal esperada en su salida para batería sea de 48V y la máxima de 57.6V (tensión máxima recomendada por el fabricante de las baterías [14]). Lo que se pretende con esto es que pueda existir un flujo de energía hacia el equipo de fuerza desde el regulador debido a que la tensión esperada por el regulador siempre será ligeramente superior a la máxima tensión que va a existir en el bus DC (porque el regulador deja de inyectar energía si se sobrepasa la máxima tensión esperada en la supuesta batería). Añadir que el regulador no va a absorber energía desde el bus DC, porque no tiene cargas conectadas en sus conectores dedicados a ello. En la figura 3.2 se muestran las entradas y salidas (junto con su propósito) del regulador .

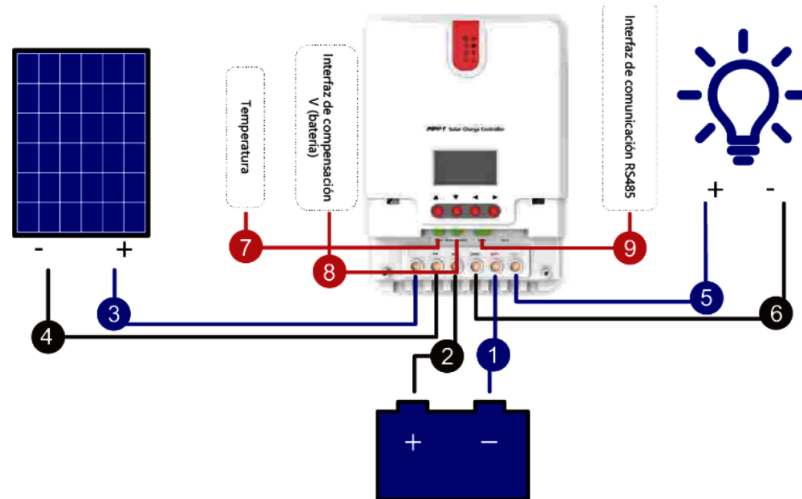


Figura 3.2: Conexiones del regulador DC-DC [13].

Asimismo, la batería está conectada al bus DC a través de un relé para controlar su conexión. Como con la configuración descrita se anula el control de corte de carga por parte del regulador DC y el equipo de fuerza, el Controlador DER (conjunto de procesos encargados de coordinar y controlar los distintos comportamientos de la microrred) abrirá o cerrará la conexión en función de la estimación del SoC que él mismo realizará. De esta manera la batería inyectará o absorberá energía del bus DC.

Por último, las 3 cargas regulables se encuentran conectadas al bus DC. Estas cargas emulan antenas menos prioritarias o equipamiento extra (como pueden ser los aires acondicionados). Las tres cargas fijas conectadas al equipo de fuerza emulan antenas prioritarias, y la energía para alimentarlas puede provenir de la red AC o del bus DC.

### ***Estándar IEC61850***

Un complemento a este trabajo es la incorporación del estándar IEC61850 al Controlador DER, para que la instalación sea un ejemplo de microrred con la capacidad de ser fácilmente extensible y poder formar así fácilmente redes de microrredes con el mismo estándar de comunicaciones y gestión de datos. Así, dentro del Controlador DER se crea un Servidor IEC61850 con el modelo estandarizado de la planta de la instalación. Este Servidor se comunicará con un Cliente IEC61850 el cual le podrá enviar comandos de actuación a más alto nivel (como estrategias de ahorro de energía o diferentes comportamientos a realizar) y adquirir información sobre la planta (medidas de sensores y variables de control).

Adjunto al Servidor se encuentra el EMS. Éste se fundamenta en una máquina de estados integrada en el Servidor, la cual hace uso de las funciones del estándar para recoger la información del modelo de la planta, actuar sobre ella según una secuenciación programada y modificar en el Servidor la información de los parámetros de control pertinentes.

Además, dentro del Controlador DER se incluyen los procesos de lectura de los sensores de la instalación y los parámetros del regulador DC, con los que el Servidor actualiza la información del modelo de la planta. En la figura 3.3 se muestra un esquema conceptual de este trabajo junto a diferentes sistemas IEC61850 con los que se pretende interactuar. Lo concerniente a este TFM se encuentra dentro del recuadro rojo discontinuo de la figura 3.3 y desarrollado en la figura 3.4, ampliando algunas interconexiones y apareciendo la estación meteorológica.

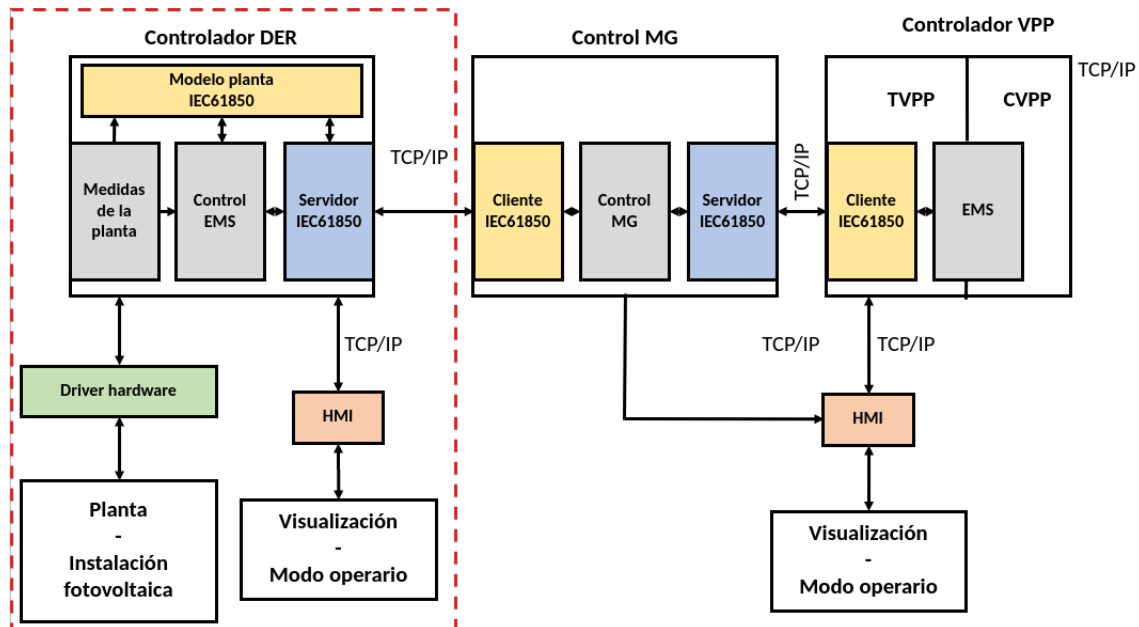


Figura 3.3: Interrelación conceptual entre sistemas IEC61850 y periféricos.

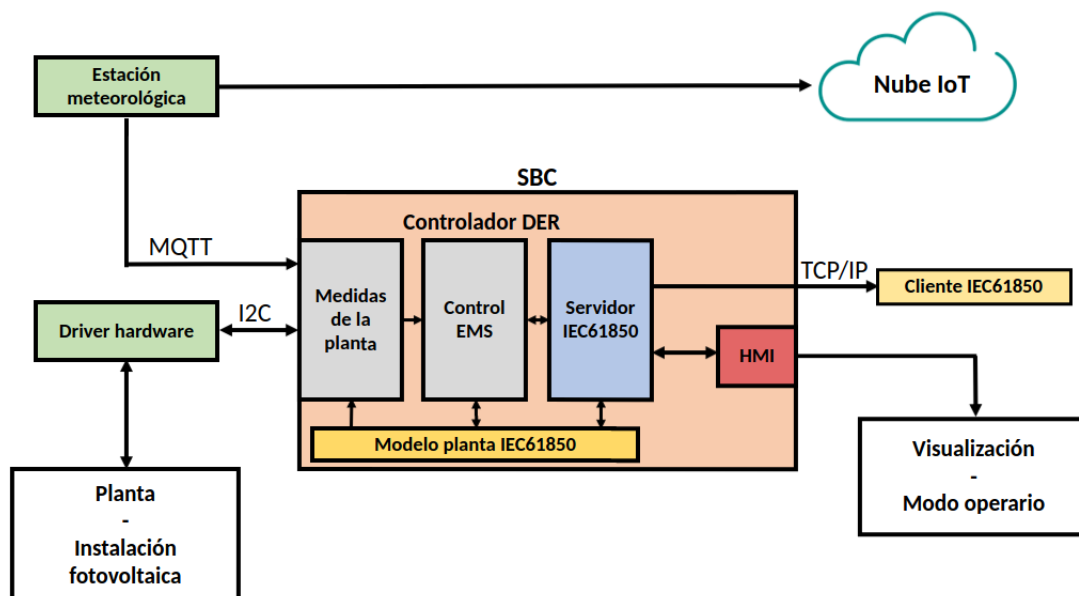


Figura 3.4: Esquema conceptual del TFM.

### *Diseño electrónico y HMI*

Por otra parte, se encuentra el driver hardware que hace de puente entre los sensores de la instalación, los distintos relés y transistores de control, y el Controlador DER. Toda la información de medidas se recogen con los procesos de lectura anteriormente mencionados, y las variables de control son enviadas al driver hardware a través de comunicación I2C.

También separado del Controlador DER se encuentra la estación meteorológica que aporta datos ambientales. Esta estación está situada junto a los módulos fotovoltaicos, alejada físicamente de la SBC que procesa el Controlador DER. La información se envía a través de MQTT al Controlador DER, compartiendo la misma red Wi-Fi.

Finalmente nos encontramos con el interfaz gráfico, montado en Node RED. Mediante esta interfaz se visualizan todos los parámetros de la instalación y también proporciona la posibilidad de controlar de forma manual la instalación.

## 3.2. Diseño electrónico

### 3.2.1. Driver Hardware

La electrónica incorporada al conjunto de la instalación actúa como driver hardware entre la planta y la SBC de control. Así, los procesos de lectura y control pueden realizar medidas eléctricas y actuar sobre los distintos elementos de la planta. Este driver hardware consta de una placa principal, dos circuitos integrados adjuntos a la SBC y dos relés externos, estando representada la disposición de los distintos elementos y sus interconexiones en la figura 3.5.

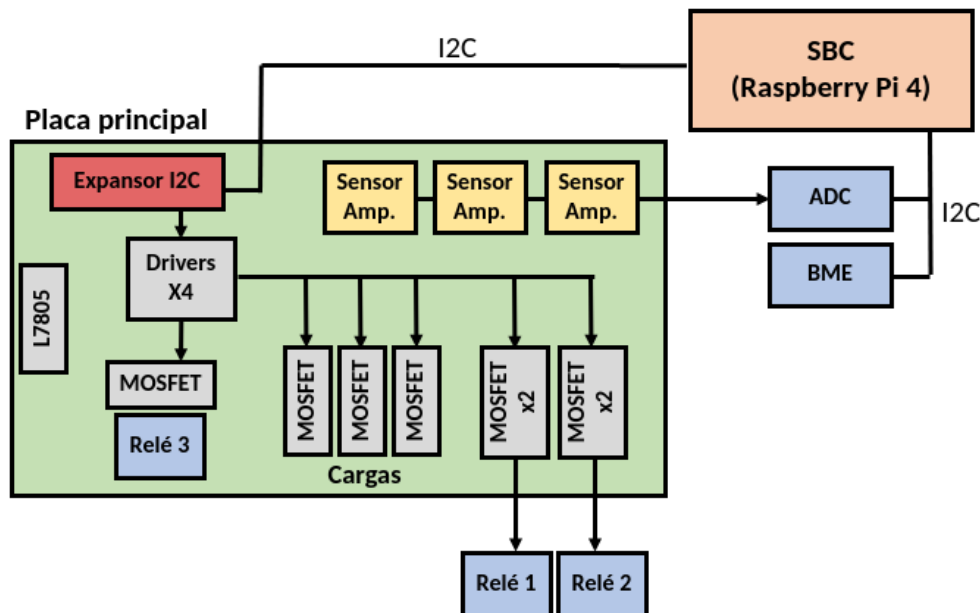


Figura 3.5: Esquema conceptual del driver hardware.

Sus funciones principales son las de realizar distintas medidas eléctricas:

- Corriente de la batería
- Corriente de las 3 cargas gestionables

- Corriente de las 3 cargas fijas
- Tensión de la batería
- Temperatura ambiente
- Humedad ambiente
- Presión atmosférica

y controlar la apertura y cierre de:

- Conexión Equipo de fuerza - Red AC
- Conexión Equipo de fuerza - Bus DC
- Conexión Batería - Bus DC
- Cierre/apertura de las 3 cargas controlables

Todas estas funcionalidades deben ser controladas a través de comunicación I2C por la SBC, por lo que los circuitos integrados encargados de la lectura deben disponer de este protocolo de comunicación y para la actuación sobre transistores y relés se añade un expansor I2C.

Otro requisito es que los sensores de corriente, transistores y relés sean capaces de manejar hasta 60A, ya que es la máxima corriente que proporciona tanto el regulador de los módulos fotovoltaicos como el equipo de fuerza. En el caso de los transistores asociados a las cargas controlables, al ser tres, esta corriente máxima se divide y la de cada transistor se reduce a 20A. Finalmente se decide sobredimensionar los componentes y tomar como corriente máxima en todos los casos 60A.

#### 3.2.1.1. Estructura

El driver hardware se compone de una placa principal (mostrada con indicaciones de sus elementos en la figura 3.6), dos circuitos integrados adjuntos a la SBC (mostrados en la figura 3.7) y dos relés externos.

La placa principal consta de:

- *Sensores de corriente.* Los sensores de corriente son 3, para las medidas de corriente de la batería, cargas controlables y cargas fijas. Para la corriente de la batería se usa un sensor bidireccional, ya que se esperan corrientes de carga y de descarga de la batería. Para la medida de las corrientes de las cargas se usan sensores unidireccionales ya que son corrientes continuas de sentido conocido. Las salidas de estos sensores son en tensión y son llevadas al ADC para su posterior conversión y lectura.
- *Divisor resistivo.* Su función es la de acondicionar la tensión de la batería para realizar su lectura mediante el ADC.
- *Expansor I2C.* Su cometido es el de capacitar el control individual de los transistores a través del puerto I2C. El expansor usado cuenta con 8 salidas digitales y consta con un registro de 8 bits donde se modifica el valor de estas salidas.
- *Regulador lineal.* Su función es la de alimentar expansor I2C a 5V.
- *Drivers.* Actúan como intermediario entre el expansor I2C y los transistores.

- *Transistores.* Son los encargados de controlar la apertura y cierre de las tres cargas controlables y los relés.

Para los transistores de las cargas se espera que atraviese por ellos una corriente máxima de 20A. Se sobredimensionan para soportar hasta 200A y disipar así mejor el calor. No obstante se decide colocar un disipador a cada uno para ayudar en este cometido.

Los transistores asociados a los relés son 5: uno para el relé de conexión AC y 2 para cada relé de enclavamiento. Estos transistores no soportarán corrientes elevadas, por lo que no dispondrán de disipadores.

- *Relé de conexión AC del equipo de fuerza.* Es el encargado de la conexión del equipo de fuerza a la red AC. Su posición será NC ya que en su funcionamiento normal no se abrirá, sólo en realización de pruebas. Se añade un diodo para la descarga de la bobina en la apertura del transistor que controla la excitación del relé.

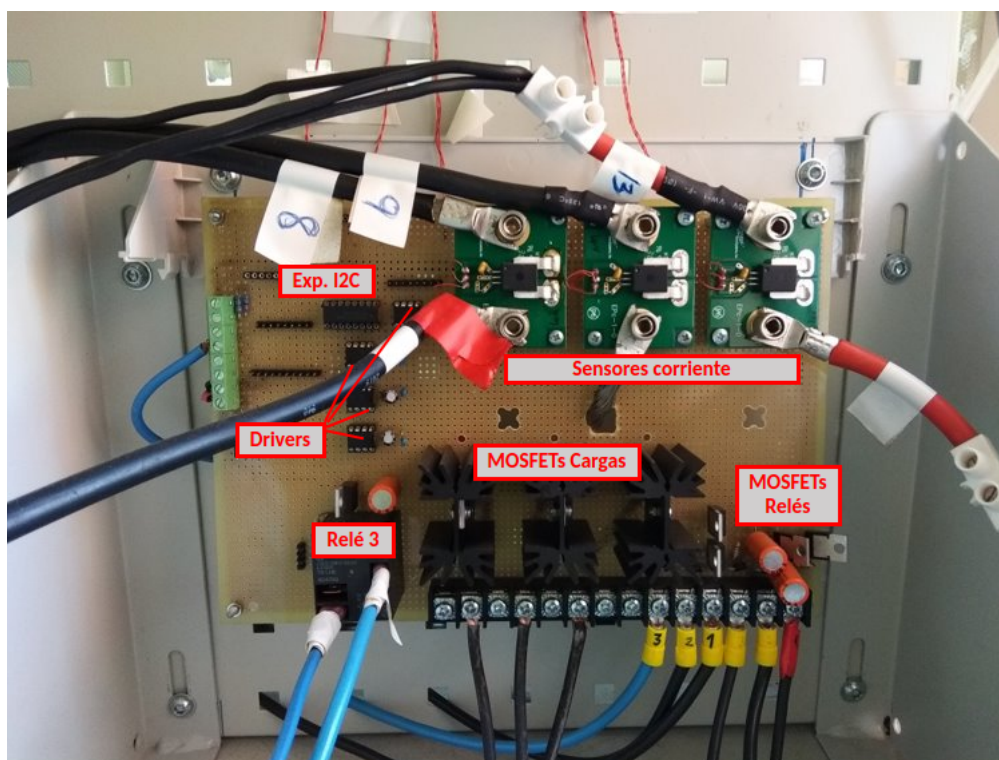


Figura 3.6: Driver hardware: placa principal.

Por otra parte se encuentran:

- *Raspberry Pi (SBC).* En lo concerniente al diseño electrónico la Raspberry Pi, esta recoge las lecturas de los distintos sensores a través del bus I2C y envía los comando de control al expansor I2C.
- *ADC.* Es el encargado de recoger la información en tensión de los tres sensores de corriente y el divisor resistivo de la placa principal. El ADC está junto a la Raspberry PI y a través del bus I2C le envía la conversión digital de las medidas.
- *Sensor de temperatura, humedad y presión.* Su función principal es la medición de la temperatura ambiente ya que es usada por el estimador SoC de las baterías. Está situada a una distancia de la Raspberry Pi suficiente para que el calor que disipa la Raspberry Pi no afecte a la medida de temperatura. Al igual que el ADC, este sensor envía la información por bus I2C a la Raspberry Pi.



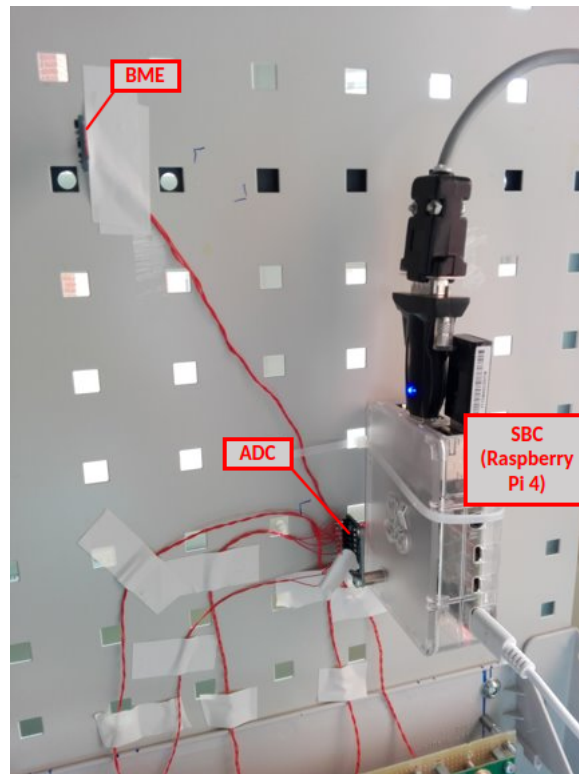


Figura 3.7: Driver hardware: SBC, ADC y BME680.

Por último se hallan dos relés de enclavamiento. Estos son los encargados de la apertura y cierre de las conexiones de la batería y el equipo de fuerza con el bus DC. Se eligen de enclavamiento para que el consumo en su control sea bajo, ya que estarán indistintamente en una posición u otra. Su control se realiza en la placa principal y mediante cableado se envían las consignas de control. Además se añade, junto a los transistores asociados, unos diodos para la descarga de las bobinas de excitación en las transiciones de conducción a apagado (figura 3.8).

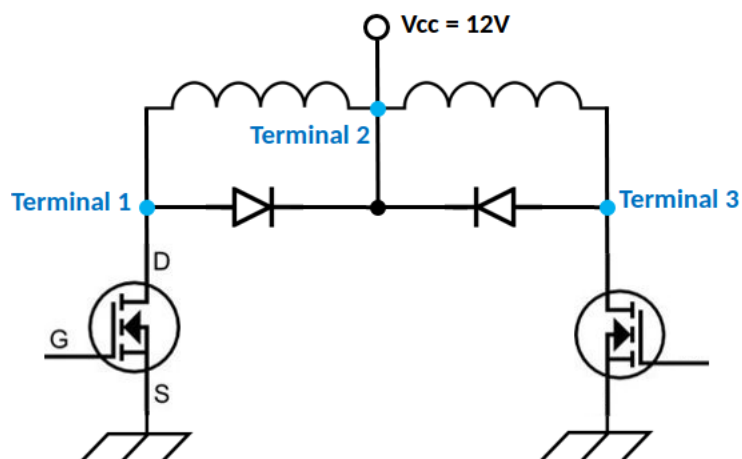


Figura 3.8: Conexionado de los relés de enclavamiento.

### 3.2.1.2. Componentes

En este apartado se enumeran los distintos componentes elegidos y sus características principales, exceptuando la Raspberry Pi. Por orden de aparición en el apartado anterior, tenemos:

■ Sensor de corriente bidireccional - ACS758LCB-100B-PFF-T [29]

- Corriente por el primario =  $\pm 100A$
- $V_{cc} = 5V$
- Sensibilidad =  $20mV/A$
- $V_{NOISE} = 6mV$  (equivalente a  $0.33A$  en la medida)
- Consumo máximo de corriente =  $I_{cc} + I_{OUTsource} + I_{OUTsink} = 13.5mA + 3mA + 1mA = 17.5mA$
- El rango de tensión de la salida es  $0V - 5V$ , coincidente con el fondo de escala configurado en el ADC que recoge la lectura.

■ Sensor de corriente unidireccional - ACS758LCB-100U-PFF-T [29]

- Corriente por el primario =  $(0 - 100)A$
- $V_{cc} = 5V$
- Sensibilidad =  $40mV/A$
- $V_{NOISE} = 12mV$  (equivalente a  $0,33A$  en la medida)
- Consumo máximo de corriente =  $I_{cc} + I_{OUTsource} + I_{OUTsink} = 13,5mA + 3mA + 1mA = 17,5mA$
- El rango de tensión de la salida es  $0V - 5V$ , coincidente con el fondo de escala configurado en el ADC que recoge la lectura.

■ Divisor resistivo -  $330K / 22K$

El divisor resistivo ( $R_1=330K\Omega$  y  $R_2=22K\Omega$ ) sirve para acondicionar la tensión de la batería para poder ser leída por el ADC. Para ello se deben cumplir tres condiciones:

- La resistencia equivalente del divisor resistivo debe ser 100 veces más pequeña que la impedancia de entrada del ADC ( $Z_e = 3M\Omega$ ), para que no afecte al acondicionamiento de la tensión a medir.

$$R_{eq} = \frac{R1 \cdot R2}{R1 + R2} = 20,625K\Omega \quad (3.1)$$

- El factor de división tiene que asegurar que la lectura se mantenga dentro del fondo de escala del ADC ( $0V - 5V$ ). La variación máxima esperada en la tensión de la batería se encontrará entre  $47V$  y  $58V$ .

$$V_{INmáx} = \frac{V_{BAT} \cdot R2}{R1 + R2} = \frac{58V}{16} = 3,625V \quad (3.2)$$

$$V_{INmín} = \frac{V_{BAT} \cdot R2}{R1 + R2} = \frac{47V}{16} = 2,938V \quad (3.3)$$

- El consumo del divisor resistivo debe ser despreciable en comparación con el normal funcionamiento de toda la instalación.

$$I_{máx} = \frac{58V}{R1 + R2} = 0,165mA \quad (3.4)$$

■ Expansor I2C - PCF8574AN [30]

- $V_{CCmáx} = 7V$
- Consumo máximo de corriente =  $100mA$

- Consumo máximo de corriente de standby =  $10\mu\text{A}$
  - 8 puertos input/output
  - La elección de este componente es debida a que el número de puertos necesarios son 8 y al bajo consumo de corriente.
- Regulador lineal - L7805 [31]
- $I_{OUTm\acute{a}x} = 1,5\text{A}$
  - $V_{INm\acute{a}x} = 35\text{V}$
  - Su cometido es el de alimentar el expansor I2C cuyo consumo máximo de corriente es de 100 mA, por lo que es más que suficiente.
- Drivers - TC4426 [32]
- $I_{OUTpeak} = 1,5\text{A}$
  - $V_{CCm\acute{a}x} = 18\text{V}$
  - $V_{OUThigh} = V_{CCm\acute{a}x} - 0,025\text{V}$
  - $V_{OUTlow} = 0,025\text{V}$
  - Consumo máximo de corriente =  $4\text{mA} \cdot 2 = 8\text{mA}$
  - El rango de alimentación es compatible con los 12V con los que se alimenta a la placa principal. Además en su salida es capaz de aportar prácticamente los valores de  $V_{CC}$  y GND, lo cual asegura una actuación correcta sobre los transistores.
- MOSFETs - CSD18536KCS 60 V N-Channel [33]
- $V_{DSm\acute{a}x} = 60\text{V}$
  - $R_{DSon} = 3\text{m}\Omega$  ( $V_{GS} = 10\text{V}$ )
  - $V_{GSth} = 1,8\text{V}$
  - $V_{GSm\acute{a}x} = \pm 20\text{V}$
  - $I_{DSm\acute{a}x} = 200\text{A}$
  - $t_{rise} = 5\text{ ns}$
  - $t_{falling} = 4\text{ns}$
  - Estos transistores son capaces de soportar 10 veces la corriente esperada por ellos (en el caso de los correspondientes a las 3 cargas controlables). Además, en ese mismo caso deberán soportar como máximo una  $V_{DS} = 58\text{V}$  que entra dentro de su margen de operación.
- Relé - JTN1S-TMP-F-DC12V [34]
- $V_{CCbobina} = 12\text{V}$
  - Corriente de contacto = 20A
  - Corriente bobina = 66,7mA
  - Terminación enchufable (conector Faston)
  - La corriente de contacto máxima que soportará es de 16A, entrando dentro del margen. El consumo de la bobina es reducido aunque en su funcionamiento normal no estará conectada.

■ ADC - ADS1115 16-bit ADC - 4 Channels [35]

- $V_{CCm\acute{a}x} = 7V$
- Consumo máximo de corriente = 0,3mA
- $V_{IN} = GND-0,3V / V_{CC}+0,3V$
- Se elige este ADC por su reducido coste, por constar de comunicación I2C y por ser uno de los más usados en aplicaciones de bajo coste. Esto último facilita su programación ya que existen multitud de librerías dedicadas a él. Además su consumo en corriente es muy reducido, el fondo de escala es suficiente para las tensiones que deberá medir, consta de 4 canales que son los necesarios para este trabajo y su resolución es más que suficiente para las medidas que va a realizar.

■ BME680 Environmental sensor [36]

- $V_{CCm\acute{a}x} = 5V$
- Consumo máximo de corriente = 16mA (En funcionamiento normal 0.4mA)
- Rango de temperatura =  $-40^{\circ}C / 85^{\circ}C$
- Rango de humedad = 0 % / 100 %
- Rango de presión = 300hPa / 1100hPa
- Se elige también este sensor por ser uno de los más utilizados, por tener comunicación I2C y por su reducido coste.

■ Relé Enclavamiento - EW60-1A3 12VDC 60A [37]

- $V_{CCbobina} = 12V$
- Corriente de contacto = 60A
- Resistencia bobina =  $50\Omega$
- La corriente de contacto máxima que soportará es de 60A, coincidiendo con la máxima soportable. El consumo en las acciones de control será de 0,24A.

### 3.2.1.3. Comunicación I2C

Toda la información recogida por los distintos sensores será proporcionada en última instancia a la Raspberry Pi 4 (SBC) a través de comunicación I2C, estando dotada con puerto I2C. De este modo se facilita la comunicación y es fácil la ampliación de sensores.

■ Expansor I2C - PCF8574AN: 0x70 (con los pines  $A_2 = A_1 = A_0 = GND$ )

Esta dirección es la asociada a la escritura en el registro. Si se quisiera leer el estado del registro la dirección sería 0x71.

■ Registro expansor I2C:

bit 0	Carga controlable 1	bit 4	Conexión Bus DC equipo de fuerza
bit 1	Carga controlable 2	bit 5	Desconexión Bus DC equipo de fuerza
bit 2	Carga controlable 3	bit 6	Conexión Bus DC batería
bit 3	Conexión AC equipo de fuerza	bit 7	Desconexión Bus DC batería

Tabla 3.1: Expansor I2C - Asignación del registro 8 bit.

- ADC - ADS1115: 0x48 (con el pin ADR = GND)
- BME680 Environmental sensor: 0x77 (con el pin CS =  $V_{CC}$ )

### 3.2.1.4. Ruido electromagnético

El ruido o interferencia electromagnética es una perturbación producida en cualquier sistema electrónico o circuito eléctrico debido a una fuente de radiación electromagnética. Este ruido puede ser interno (intrínseco) si es generado internamente en los dispositivos electrónicos a causa de diferentes fenómenos físicos, o externo (interferencias) si son señales acopladas por fuentes externas naturales o artificiales (otros sistemas de índole eléctrica).

Encontramos además dos tipos de acoplamiento en función del camino de transmisión del ruido o interferencia: acoplamiento conducido si la interferencia llega a través de un camino eléctrico, o acoplamiento radiado si la interferencia llega a través de una camino no eléctrico mediante una onda electromagnética. Estos acoplamientos se representan en la figura 3.9. También podemos encontrar el acoplamiento mixto, hallando aportaciones de ambos [38].

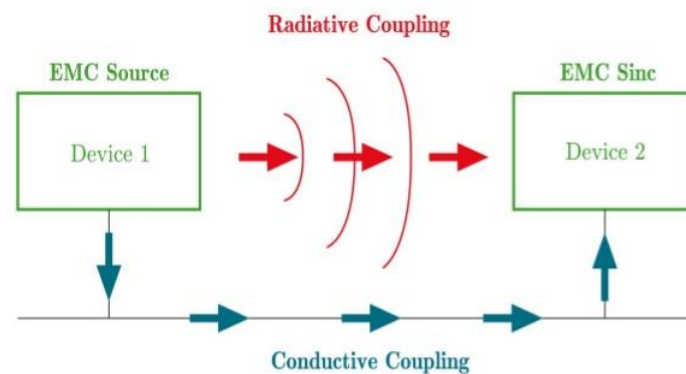


Figura 3.9: Ruido electromagnético: tipos de acoplamiento [39].

#### *Versión inicial*

Se construye una versión inicial del driver hardware para el control de la planta (figura 3.10). En esta primera versión o prototipo no se tiene en cuenta la disposición de los diferentes elementos atendiendo a interferencias electromagnéticas.

Con este prototipo inicial se comprueba el buen funcionamiento de todos los elementos en el laboratorio, pero en las pruebas de funcionamiento real junto a la instalación se observan ruidos en las medidas, medidas incoherentes, ocasional pérdida de información e incluso pérdida de la comunicación con el sensor BME680.

Se observa que estos fenómenos tienen lugar cuando se cierra alguna de las cargas controlables, pasando por cada una de ellas alrededor de 17A DC. Se estudia entonces la disposición del cable que lleva esta corriente, en concreto el que conecta la masa del bus DC y el sensor de corriente. Este se encuentra muy próximo a los cables que llevan la información de los sensores de corriente hacia el ADC (inicialmente en la placa principal), al sensor BME 680 (inicialmente en la placa principal) y al bus I2C. Se determina así que puede estar produciéndose un acoplo inductivo en estos elementos a través de los cables de alimentación, apareciendo un ruido en tensión por la generación de un campo magnético cuando varía la corriente en la apertura o cierre de alguna carga.

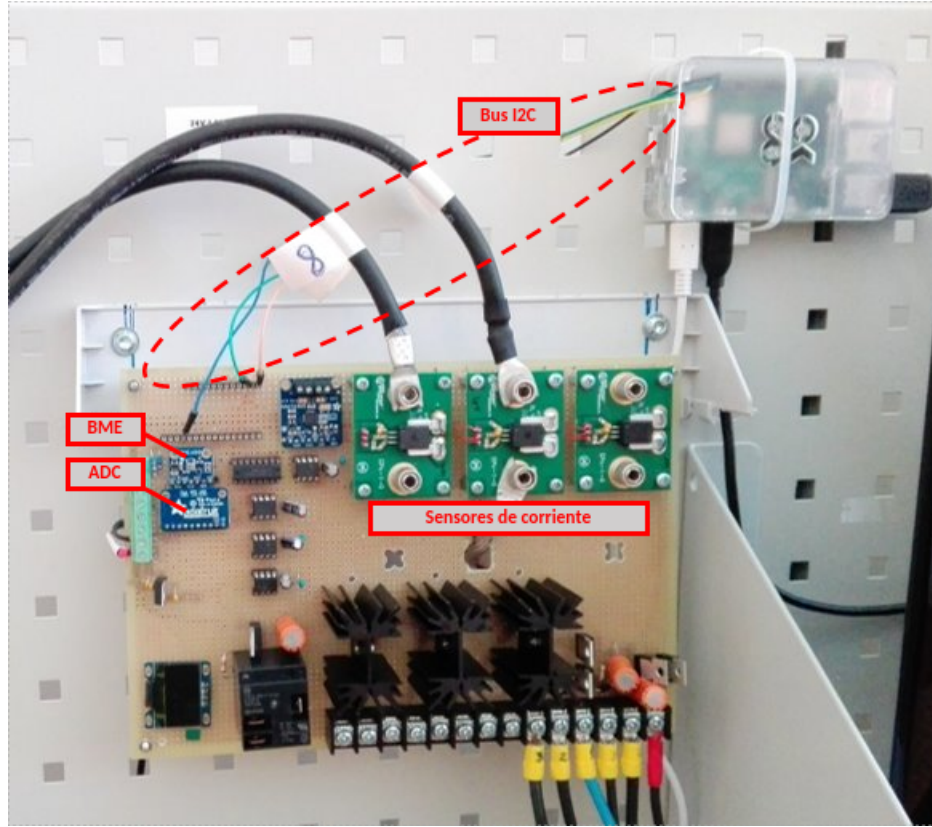


Figura 3.10: Disposición inicial del driver hardware.

### *Estudio de las interferencias*

Se parte de la base de que la interferencia es inductiva y radiada en modo diferencial por la disposición de los distintos elementos del driver hardware y la planta.

#### Interferencia radiada

En primer lugar se estudia la posible interferencia inductiva y radiada. La variación momentánea de corriente de 0A a 17A (en el mejor caso), de 0A a 34A (caso intermedio) y de 0A a 51A (en el peor) genera un campo magnético que induce una tensión ruido en los elementos.

Para calcular este ruido, se puede estimar que la distancia ( $R$ ) del cable respecto área de interferencia de los cables alimentación es aproximadamente de 1cm y el área es aproximadamente de 25cm<sup>2</sup>. La densidad de flujo magnético (en el caso de 34A de variación) se calcula como:

$$B(t) = B_0 \cdot e^{j\omega t} \quad (3.5)$$

$$|B(t)|_{m\acute{a}x} = \frac{\mu_0 \cdot i}{2\pi \cdot R} = B = 6,8 \cdot 10^{-4} T \quad (3.6)$$

$$\mu_0 = 4\pi \cdot 10^{-7} [T \cdot m/A]$$

Con la densidad de flujo magnético podemos calcular el flujo magnético, considerando que el ángulo formado por el vector unitario del campo magnético y el vector normal a la superficie del área es  $\theta = 0^\circ$ , y el tiempo de subida del transistor  $t_{rise} = 5ms$  (mejor caso) es el 80 % del tiempo de pico de la respuesta transitoria [40] (suponiendo una respuesta críticamente amortiguada ( $\xi = 1,0$ )).

$$t_p = \frac{t_{rise}}{0,8} = 6,25ms \quad (3.7)$$

$$\omega_n = \frac{2\pi}{T} = \frac{\pi}{t_p} = 160\pi rad/s \quad (3.8)$$

$$\phi = B_0 \cdot e^{j\omega t} \cdot A \cdot \cos\theta \quad (3.9)$$

Por último la tensión inducida se calcula como [38]:

$$|V_m| = \left| -\frac{d}{dt}\phi \right| = \omega \cdot B \cdot A \cdot \cos\theta = 21,36mV \quad (3.10)$$

Una tensión ruido de 21mV es suficiente para afectar a la información transmitida de los sensores de corriente, ya que la sensibilidad de los sensores unidireccionales es de 40mV/A y la del bidireccional de 20mV/A. Esto se traduce en un posible error en la medida de 0,5A a 1A.

Además, el integrado BME680 está optimizado para una tensión de alimentación interna ("*Supply Voltage Internal Domains*") de 1.8V y no tolera un rizado superior a 50mVpp [36], por lo que este ruido puede afectar a su funcionamiento.

Finalmente, si consideramos el peor caso en el transitorio de cierre del transistor (sólo cambiando  $t_{fall}=1ms$ ) obtenemos:

$$t_p = \frac{t_{fall}}{0,8} = 1,25ms \quad (3.11)$$

$$\omega_n = \frac{\pi}{t_p} = 800\pi rad/s \quad (3.12)$$

$$|V_m| = \left| -\frac{d}{dt}\phi \right| = \omega \cdot B \cdot A \cdot \cos\theta = 106,81mV \quad (3.13)$$

Por lo que el ruido introducido de 106mV puede suponer un error de medida de los sensores de corriente de hasta 5A y alcanzar el rizado máximo permitido en la alimentación del BME680.

### Interferencia conducida

En segundo lugar se estudia si la interferencia puede ser conducida en modo común a través de las masas del sistema.

Siguiendo la figura 3.11, observamos que a través de  $R_7$  se puede introducir un error igual a  $V_m - V_{bat(-)}$  (ecuación 3.14) en la medida de tensión de la batería, ya que la referencia de la masa de la placa principal está tomada en  $P_m$ .

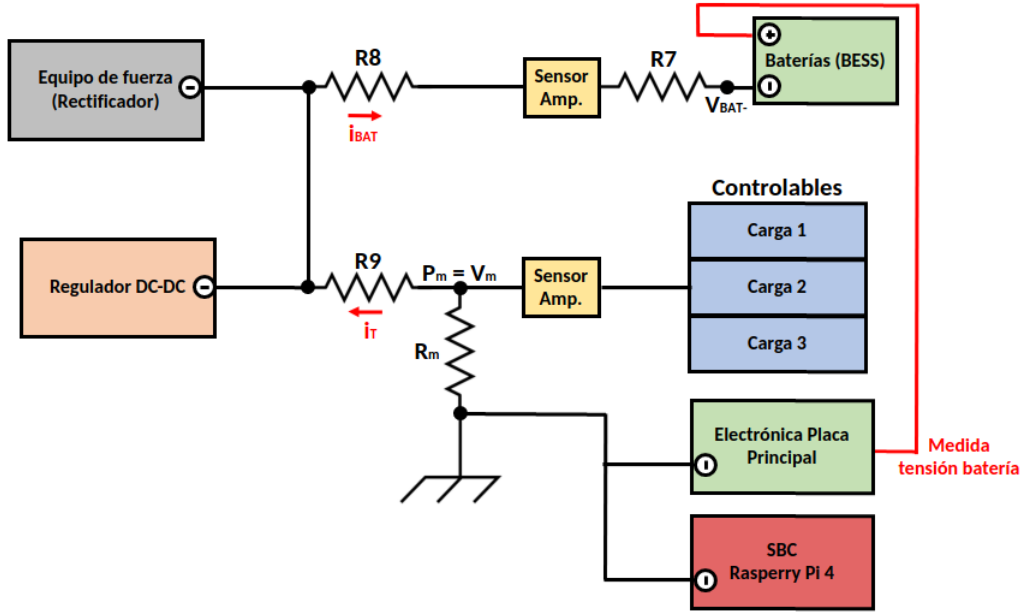


Figura 3.11: Estudio del ruido conducido por la masa del sistema.

$$V_m - V_{bat-} = R_9 \cdot (i_{cargas} + i_{bat}) + (R_7 + R_8) \cdot i_{bat} \quad (3.14)$$

Además, la corriente que circula por  $R_9$  puede producir una subida de tensión en  $P_m$ , provocando una corriente a través del divisor resistivo conformado  $R_m$  y  $R_A$  e introduciendo así ruido en la alimentación de la electrónica de la placa principal.

Para conocer la magnitud de la aportación de estas resistencias se miden:

$$R_7 = R_8 = R_9 = 0,0\Omega$$

$$R_m = 0,1\Omega$$

$$R_A = 0,0\Omega$$

Por lo que la aportación de  $R_7$ ,  $R_8$  y  $R_9$  se considera despreciable, y la aportación del divisor resistivo ( $R_m/R_A$ ) ya que  $R_m \gg R_A$  (figura 3.12).

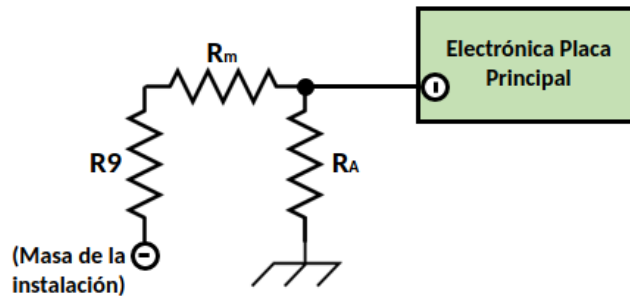


Figura 3.12: Estudio del ruido conducido por el divisor resistivo ( $R_m/R_A$ ).

Se puede concluir finalmente que las interferencias son intrasistema producidas por acoplo inductivo y radiado en modo diferencial.



### Versión final

Tras concluir en que el ruido es producido por acoplo inductivo y radiado en modo diferencial (y que no se puede disminuir el campo magnético en la fuente) se procede a reducir el ruido acoplado según [38]:

- *Alejar los integrados sensibles de la fuente de interferencias*

El integrado BME680 y el ADS1115 se colocan fuera de la placa principal, y la Raspberry Pi 4 se aleja aún un poco más de la placa principal. El ADC pasa ahora a estar junto a la Raspberry Pi 4 (figura 3.16).

- *Orientar el plano del circuito paralelo al campo magnético*

Tanto los circuitos del BME680, el ADS1115, la Raspberry Pi 4 como los cables de transmisión de medidas y el bus I2C se colocan de forma perpendicular a los cables que generan el campo magnético (figura 3.16). Así se consigue que  $\cos\theta = 0$  ya que la polarización del campo magnético es conocida (mostrado conceptualmente en la figura 3.13).

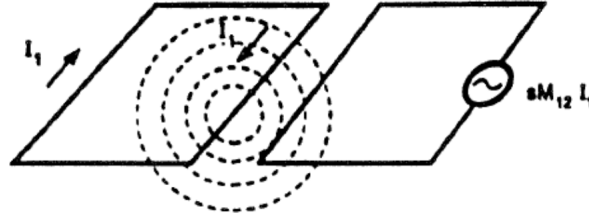


Figura 3.13: Área de bucle con máxima interferencia ( $\cos\theta = 90$ ) [38].

- *Reducir área de las transmisiones*

Los cables que llevan la información y la alimentación de los sensores de corriente se juntan entre ellos lo máximo posible. A los cables del bus I2C se les realiza la misma operación.

- *Par trenzado en transmisiones*

A estos mismos cables se les realiza un trenzado para que las tensiones inducidas en los bucles consecutivos tiendan a anularse estadísticamente. En la figura 3.14 se muestra un detalle de estos cables.

- *Reestructuración de las masas*

Para no crear grandes áreas de captación de interferencias con los cables de masa, se reestructuran según se indica en la figura 3.15.

#### 3.2.1.5. Alimentación

Como apartado final del diseño electrónico se estudian las distintas alimentaciones del conjunto.

En primer lugar la Raspberry Pi 4 alimenta al integrado BME680, al ADS1115 y a los sensores de corriente. La fuente de alimentación de la Raspberry Pi 4 es capaz de proporcionar 3A [41], pero hay que descontar el consumo máximo de los USB (1,1A). Así, el consumo total de la Raspberry Pi 4 (ecuación 3.15) es viable.

$$i_{consumo} = i_{USBm\acute{a}x} + i_{sensoresCCm\acute{a}x} + i_{ADCm\acute{a}x} + i_{BME680m\acute{a}x}$$

$$i_{consumo} = 1,1A + 60mA + 0,3mA + 16mA = 1,1763A$$
(3.15)

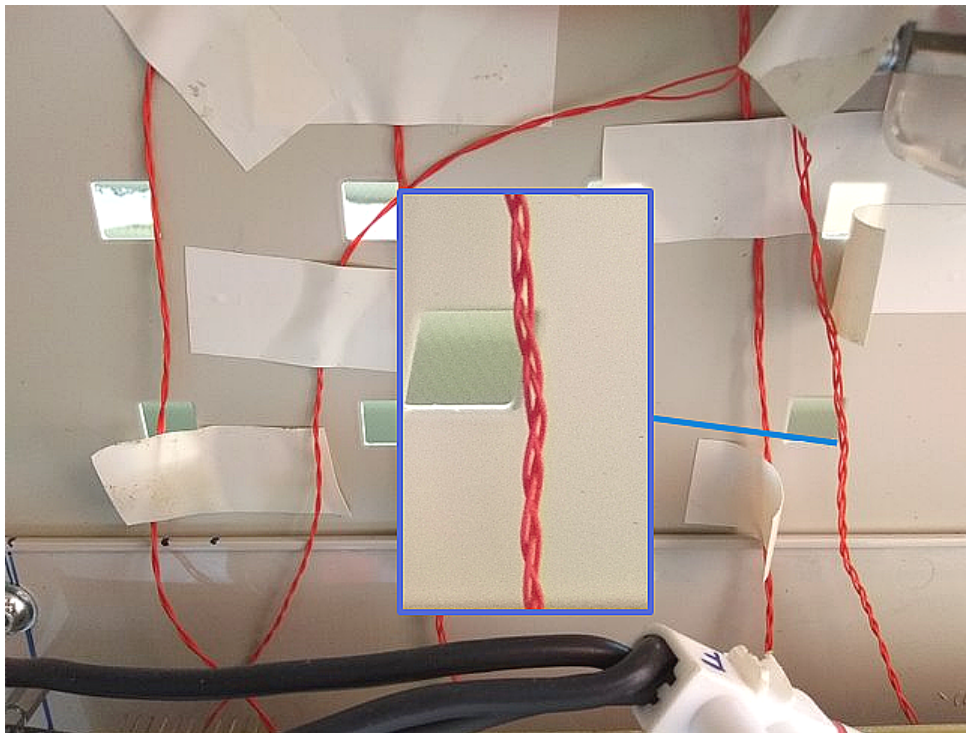


Figura 3.14: Cables trenzados. Transmisión de datos y alimentación.

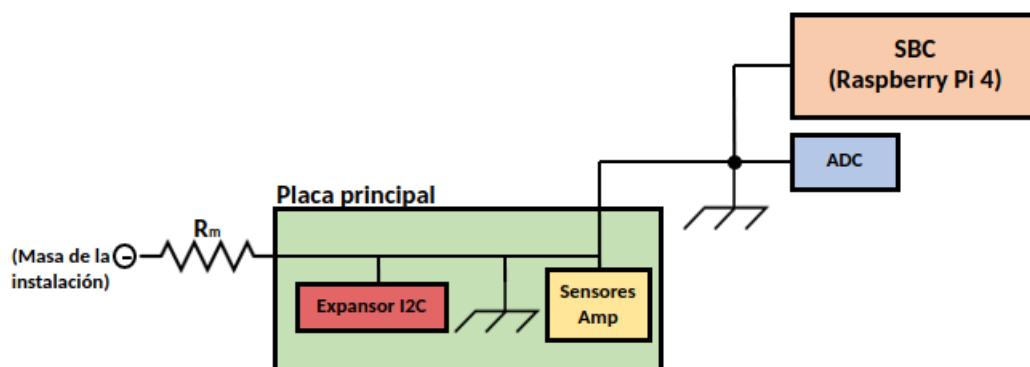


Figura 3.15: Disposición de las masas del driver hardware.

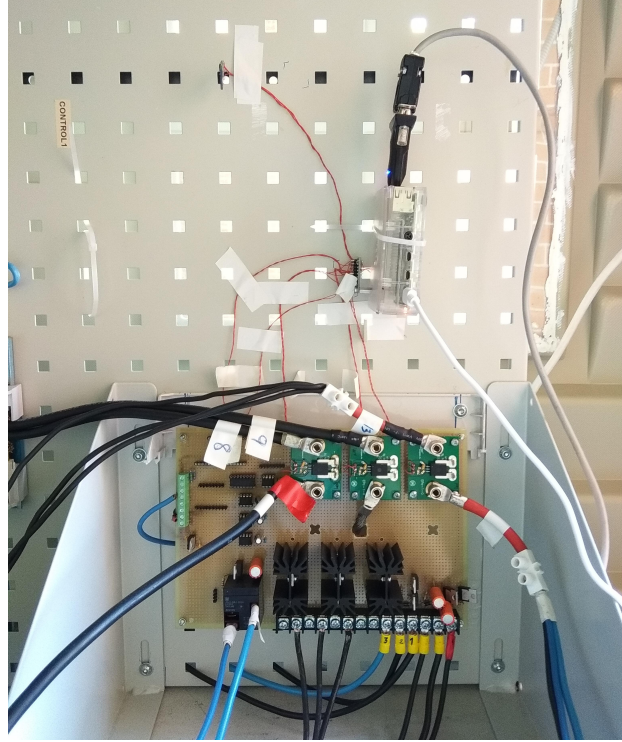


Figura 3.16: Driver hardware: versión final.

En segundo lugar la placa principal es alimentada por una fuente de 12V y 1A. Los elementos a alimentar son el expansor I2C, los drivers y las bobinas de los relés. Así, el consumo total de la placa principal es viable.

$$i_{consumo} = i_{expI2C} + i_{drivers} + i_{relé1} + i_{relé2} + i_{relé3} \quad (3.16)$$

$$i_{consumo} = 100mA + 8mA \cdot 4 + 66,7mA + 0,24A + 0,24A = 0,6787A$$

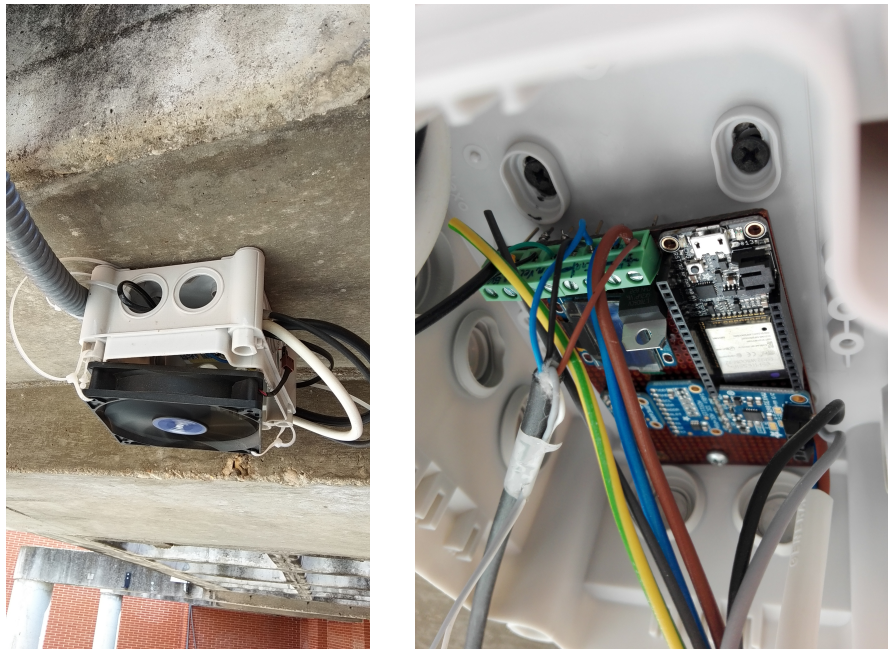
Finalmente se añaden unos condensadores electrolíticos de  $200\mu F$  para proporcionar los picos de 0.24A en los cambios de estado de los relés de enclavamiento. Además se reduce así la introducción de un ruido conducido a través de la alimentación en esas demandas momentáneas de corriente.

### 3.2.2. Estación meteorológica

Dentro de la microrred de este proyecto se incluye una estación meteorológica. Esta estación está situada junto a los módulos fotovoltaicos, alejada físicamente de la SBC que procesa el Controlador DER (mostrándose su instalación en las figuras 3.17a y 3.17). Su objetivo principal es aportar datos meteorológicos para un uso directo en el EMS o para su procesamiento a más alto nivel y generar estrategias en función de la predicción meteorológica.

La información recogida por la estación se envía a través de MQTT al Controlador DER. Un nodo en ROS hace de puente entre MQTT y ROS, publicando los datos de la estación meteorológica para que los demás nodos de ROS puedan suscribirse y disponer de dicha información.

MQTT es un protocolo estándar de comunicaciones de OASIS pensado para su uso en IoT. Está diseñado como un transporte de mensajería publicador/subscriptor extremadamente ligero, lo que es



(a) Electrónica de adquisición con sistema de ventilación.

(b) Electrónica de adquisición descubierta.

Figura 3.17: Estación meteorológica.

ideal para conectar dispositivos remotos simples y con un ancho de banda mínimo. Hoy en día MQTT es usado para una gran variedad de aplicaciones en la industria, como en la automoción, telecomunicaciones, gas y petróleo, etc. [42]. En la figura 3.18 se muestra su logotipo.



Figura 3.18: Logotipo MQTT [43].

Se decide realizar la comunicación por MQTT (en vez de directamente en ROS) porque la estación meteorológica dispone de su propio microprocesador (ESP32) y dada su simplicidad en comparación con la Raspberry Pi 4 y los procesos que alberga, se decide no incluir ROS en él. De esta forma la estación meteorológica mantiene su filosofía de simplicidad y bajo consumo de recursos.

### 3.2.2.1. Hardware

La estación meteorológica se divide en dos partes: sensores y electrónica de adquisición. Los sensores externos a la electrónica recogerán datos de irradiancia, velocidad del viento, dirección del viento y temperatura de los módulos fotovoltaicos. La electrónica cuenta con un microprocesador Feather ESP32, electrónica de acondicionamiento, un ADC, un módulo BME680 para la recogida de la temperatura ambiente, humedad y presión atmosférica, y un ventilador. En la figura 3.19 se muestra la distribución de los diferentes elementos.

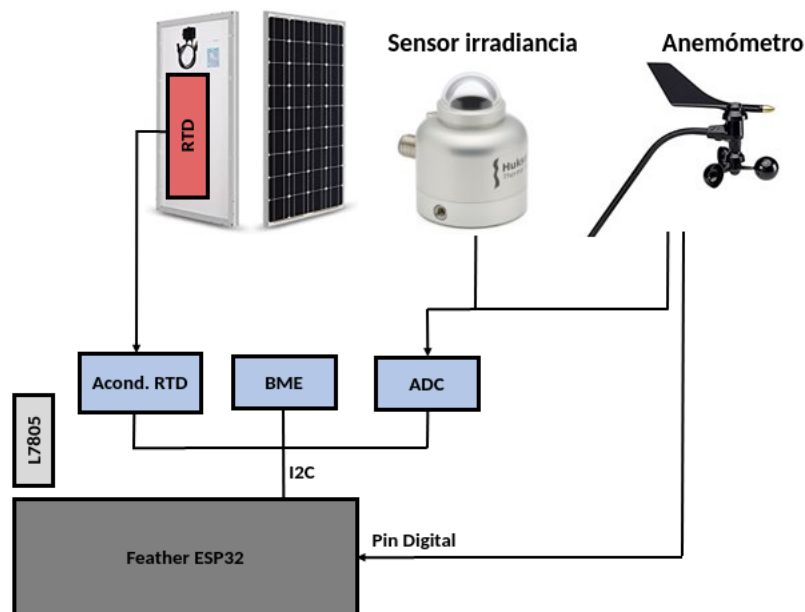


Figura 3.19: Distribución conceptual de los componentes de la estación meteorológica.

#### ■ Sensor de irradiancia SR05 [44]

SR05 series es uno de los pirómetros más asequibles cumpliendo con la normativa ISO 9060. Es ideal para una medida general de la irradiancia solar en estaciones meteorológicas y sistemas de monitorización fotovoltaica. La información puede ser recogida de forma digital (a través de RS-485) o analógica (en tensión). Se encuentra junto a los módulos fotovoltaicos (figura 3.20a), de tal forma que pueda captar la misma irradiancia que estos.

La información se obtiene de forma analógica por tensión, siendo recogida por el ADC. Esta tensión se encuentra entre 0-1V ( $0-1000\text{W}/\text{m}^2$ ), pudiendo llegar hasta 2V que corresponderían con  $2000\text{W}/\text{m}^2$ . El sensor incorpora la electrónica de acondicionamiento necesaria, por lo que se puede recoger la tensión directamente con el ADC.

#### ■ Anemómetro [45]

El anemómetro incluye sensores para la velocidad del viento y su dirección. Los rangos de trabajo son de 1 a 322km/h para la velocidad del viento con una resolución de 1km/h, y de  $0^\circ$  a  $360^\circ$  para la dirección del viento con una resolución de  $1^\circ$ . Además lleva a cabo un filtrado digital aplicado a la medida de la dirección del viento. El anemómetro se encuentra cerca de los módulos fotovoltaicos, aunque instalado en una cornisa para poder captar adecuadamente el viento (figura 3.20b).

La veleta tiene una potenciómetro lineal de  $20\text{k}\Omega$  para poder medir la dirección del viento. La salida de este potenciómetro es llevada al ADC. Por otra parte, el anemómetro tiene un *reed switch* o relé *reed* (interruptor activado por campo magnético) montado cerca del eje de rotación. Este interruptor es activado una vez por cada revolución del eje. Para calcular la velocidad del viento ha de aplicarse una fórmula proporcionada en la hoja de características [45]. El acondicionamiento para la adquisición de los datos del anemómetro se muestra en la figura 3.21.





(a) Pirómetro SR05.



(b) Anemómetro y veleta de la estación meteorológica.

Figura 3.20: Estación meteorológica.

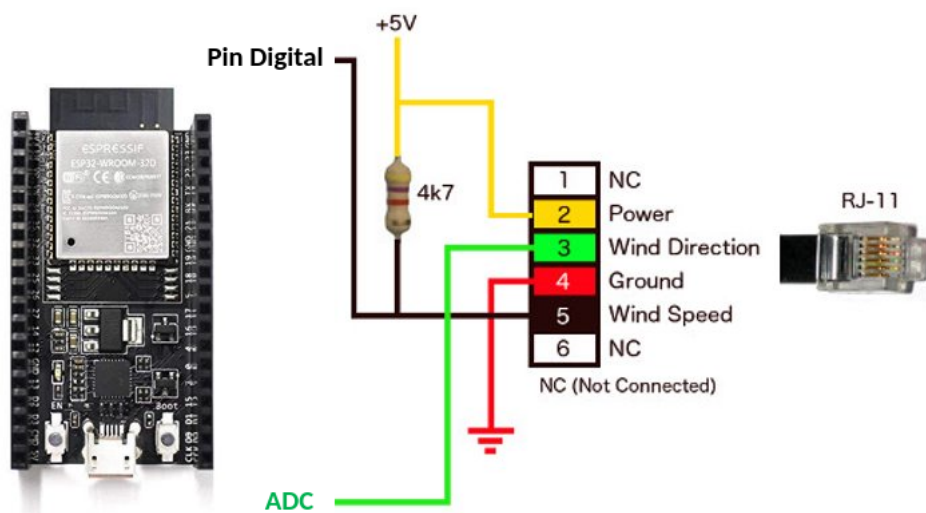


Figura 3.21: Acondicionamiento del anemómetro y veleta.

- RTD y módulo MAX31865 [46]

Una RTD (Resistance Temperature Detector) es un sensor de temperatura basado en la variación de la resistencia de un conductor con la temperatura. La variación de su resistencia es lineal, siendo su coeficiente de temperatura positivo.

El módulo MAX31865 cuenta con la electrónica de acondicionamiento necesaria para la medida de una RTD. Cuenta con la posibilidad de conexión a 2 o 4 hilos. Se elige conexión a 4 hilos ya que la medida de la resistencia es mucho más precisa y elimina posibles ruidos de offset e interferencias. La RTD usada es un pt100 (resistencia nominal de  $100\Omega$ ) y es colocada en la parte posterior de uno de los módulos fotovoltaicos (figura 3.22).

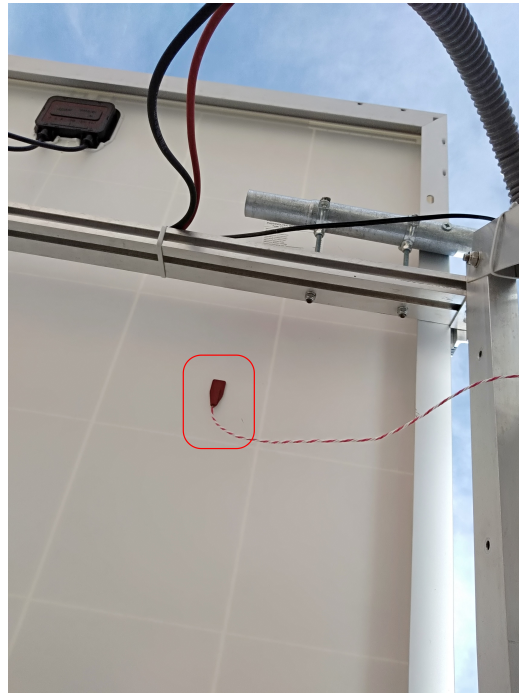


Figura 3.22: Montaje de la RTD en el módulo fotovoltaico.

- Microcontrolador Feather ESP32 [47]

Se usa principalmente el microcontrolador Feather ESP32 por su capacidad de comunicación por Wi-Fi, por tener suficientes pines de propósito general (GPIO) y por sus buenas prestaciones a nivel software. Este microprocesador lee por I2C la información del ADC, del sensor ambiental y del módulo MAX31865 del RTD. Además, la velocidad del viento proporcionada por el anemómetro es obtenida por un GPIO digital.

- ADS1115 [35]

El ADC ADS1115 es el encargado de recoger la información suministrada por el sensor de irradiancia y el anemómetro de la dirección del viento. Las tensiones digitalizadas las envía a través de comunicación I2C al microcontrolador.

- BME680 [36]

El módulo BME680 recoge y procesa a las unidades necesarias la temperatura ambiente ( $^{\circ}\text{C}$ ), humedad (%) y presión atmosférica (hPa). Esta información la envía al microcontrolador a través de comunicación I2C.

Para la alimentación de toda la electrónica se usa un regulador lineal L7805 (5V), capaz de suministrar 1,5A [31]. Las corrientes máximas de los diferentes elementos no exceden la máxima del regulador:

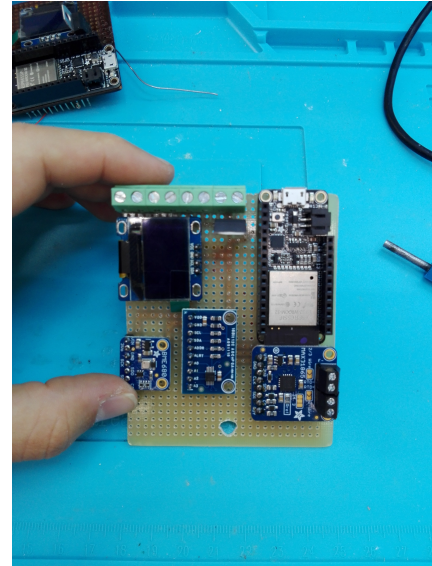
$$i_{consumo} = i_{s.irrad.} + i_{anemt.} + i_{RTD} + i_{ESP32max} + i_{ADC} + i_{BME} \quad (3.17)$$

$$i_{consumo} = 40mA + 5mA + 3,5mA + 1A + 0,3mA + 16mA = 1,0648A$$

Junto con toda la electrónica se instala un ventilador con dos funciones: refrigerar el L7805 y mantener a temperatura ambiente el sensor BME680. En la primera de ellas se debe tener en consideración que si se aumenta la tensión de entrada del L7805 (sin disipador) por encima de los 12V, éste va a disipar alrededor de 5W, suponiendo un estrés para el componente y una fuente de calor que puede afectar a las medidas tomadas por el módulo BME680. La segunda función es mantener al módulo BME680 a temperatura ambiente real y no falsear así la medida de temperatura. La instalación de la electrónica junto con el ventilador se muestran en la figura 3.23a y la electrónica en detalle en la figura 3.23b.



(a) Sistema de ventilación.



(b) Electrónica de adquisición (detalle).

Figura 3.23: Electrónica de adquisición: montaje.

### 3.2.2.2. Software

El procesador ESP32 está preparado para albergar un sistema operativo en tiempo real ampliamente conocido FreeRTOS. FreeRTOS es un sistema operativo de tiempo real (RTOS) líder en el mercado actual, diseñado para microcontroladores y pequeños microprocesadores. Éste es distribuido gratuitamente por el MIT. FreeRTOS (logotipo mostrado en figura 3.24) incluye un kernel y un conjunto de librerías en constante crecimiento. Estas librerías son adecuadas para su uso en todos los sectores de la industria.



Figura 3.24: Logotipo FreeRTOS [48].

Se elige FreeRTOS para tener un mayor control en las diferentes tareas a realizar y por tener muchas librerías y ejemplos disponibles que aceleran el proceso de desarrollo.



La estructura del programa utilizado en FreeRTOS consta de dos tareas, cada una de ellas alojada en un núcleo distintos (figura 3.25). Esto se hace así porque en pruebas iniciales de transmisión de datos la conexión Wi-Fi se bloqueaba, y el núcleo restante podía detectarlo y reiniciar el sistema. Las dos tareas utilizadas son:

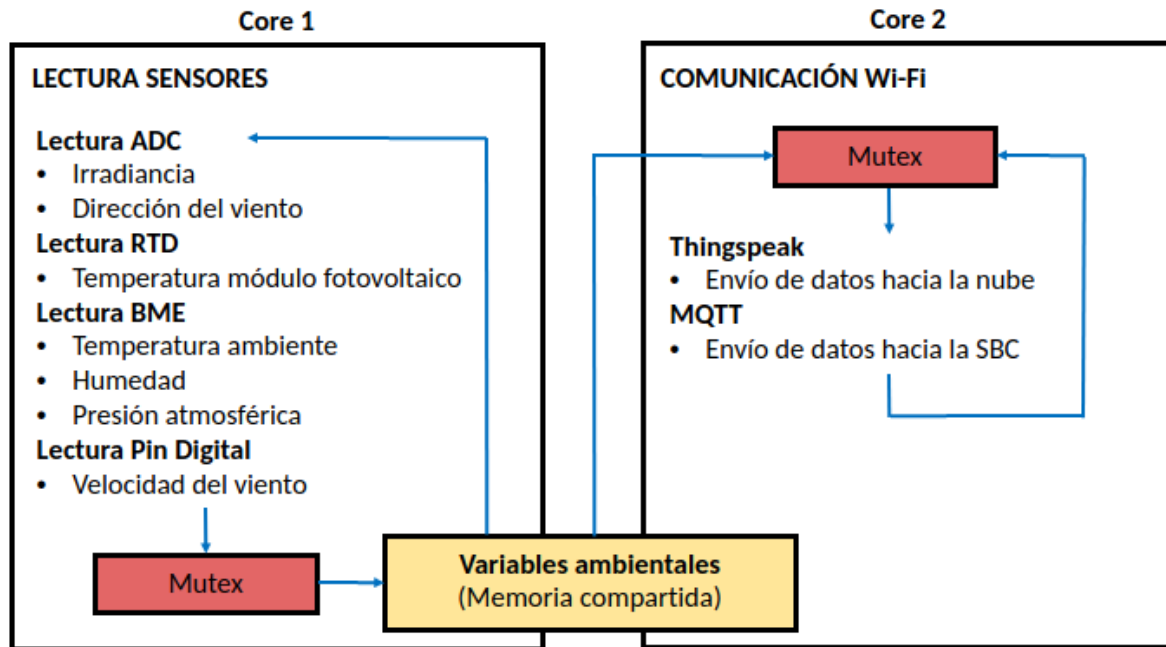


Figura 3.25: Esquema conceptual del programa de la estación meteorológica.

#### ■ Lectura de sensores

Esta tarea se ejecuta cada 1 minuto, recogiendo toda la información de los sensores a través de comunicación I2C. Las tensiones adquiridas del ADC son procesadas para obtener la información final de la irradiancia y la dirección del viento. La información de la temperatura de los módulos fotovoltaicos es procesada con una librería específica del módulo MAX31865.

Una vez obtenida toda la información, mediante un mótex se guarda de forma segura en variables globales para se usada por la otra tarea.

#### ■ Comunicación MQTT y Thingspeak

Esta tarea recoge la información en las variables globales mencionadas anteriormente también con la ayuda del mótex para evitar un conflicto en el acceso a memoria. La información de los sensores es enviada a través de MQTT hacia la SBC (Raspberry Pi 4) y hacia la nube de Matlab Thingspeak, simulando una futura funcionalidad de procesamiento a alto nivel de la microrred.

Este procesamiento estaría enfocado a realizar predicciones meteorológicas en la nube y después variar el comportamiento de la microrred. En este proyecto solamente se envían estos datos a Thingspeak para su visualización y posterior recolección. Otros trabajos de investigación relacionados a este TFM usan actualmente esta información para ser llevados a cabo.



y convenios para poder construir y ejecutar código entre múltiples ordenadores y/o procesadores. Su logotipo es el mostrado en la figura 3.27.



Figura 3.27: Logotipo ROS [51].

ROS es una red de procesos (potencialmente distribuidos a lo largo de diferentes máquinas) que están acoplados usando la infraestructura de comunicaciones de ROS. Presenta diferentes estilos de comunicación, incluyendo transmisión asíncrona de datos a través de tópicos, comunicación RPS a través de servicios y almacenamiento de datos.

Se decide utilizar ROS ya que aporta una base de comunicaciones estable entre los procesos del Controlador DER. Además, al ser una plataforma de software libre con una amplia comunidad en el ámbito de la ingeniería que la mantiene y con gran documentación disponible, sigue la filosofía de este trabajo. Los distintos procesos del Controlador DER se organizan en distintos nodos de ROS. Cada nodo (o proceso) publica información a través de tópicos (o *topics*) y adquiere información de otros nodos a través de la subscripción a los mismos.

En la versión inicial de este controlador los procesos se programaron en un solo proceso de Linux, separando las distintas partes en hilos. Tampoco se usó ROS como base para las comunicaciones, utilizando en su lugar comunicación de más bajo nivel por *sockets*. El resultado que se obtuvo fue desfavorable, encontrando fallos eventuales en las comunicaciones y errores en la generación o destrucción de los distintos hilos.

La totalidad del Controlador DER está alojada en una Raspberry Pi 4 con el sistema operativo "*Raspberry Pi OS*" (anteriormente denominado "*Raspbian*"). Esta SBC es ideal para el propósito de este trabajo, ya que cuenta con un nuevo procesador Broadcom que es hasta tres veces más eficiente que el de la Raspberry Pi 3. Además, esta versión incorpora el USB 3.0, y el puerto Ethernet no está limitado a 300 Mbps, por lo que las comunicaciones con Clientes IEC61850 no estarán limitadas.

#### 3.3.1.1. Nivel 1 - Interacción física

Como se introducía anteriormente, el primer nivel está referido a la lectura de sensores y recogida de parámetros de la microrred. Los procesos de este nivel realizan operaciones de comunicación I2C, Modbus, MQTT y procesamiento de la información adquirida de los sensores.

##### *Lectura de sensores*

Este proceso se encarga de la lectura del ADC (ADS1115) y del sensor ambiental (BME680) mediante comunicación I2C, y de determinar el estado de carga de las baterías (SoC). En la figura 3.28 se muestra un diagrama de la secuenciación del proceso al completo.

##### Inicialización y lectura I2C

En primer lugar se declaran todas las constantes a utilizar en el procesado de los datos de los sensores. Destacar que determinar los distintos *offsets* de los sensores de corriente es crucial para su buena medida a través del ADC, ya que tienen una sensibilidad de 40mV/A y 20mV/A. Se inicializan también todas las funciones necesarias para la comunicación I2C con el ADC ADS1115 y el sensor BME680.

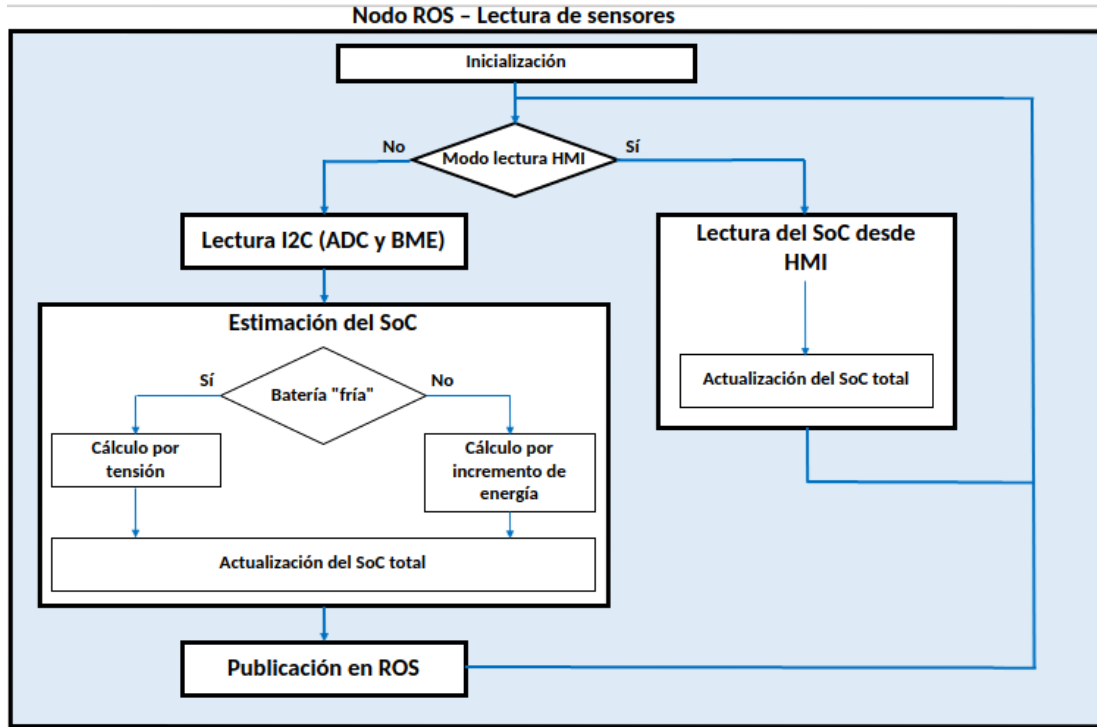


Figura 3.28: Esquema conceptual del nodo ROS lectura de sensores.

Una vez realizada la declaración inicial, se procede con la función principal de este nodo. Es una función "Callback" de ROS, asociada a un temporizador que cada 3 segundos ejecuta la función. La primera vez que se ejecuta inicializa la comunicación I2C con los dos sensores ADS1115 y BME680. Tras esto, se adquieren primero los datos del sensor BME680 ya en las unidades necesarias (temperatura (°C), humedad (%) y presión atmosférica (hPa)) y después se adquieren las tensiones de los cuatro canales del ADC.

Con las medidas en tensión se calculan la tensión de la batería (divisor resistivo) y las corrientes de batería, cargas fijas y cargas controlables. Con la tensión y corriente de la batería se pasa a estimar el SoC.

#### Estimación del SoC

Para la estimación del SoC se realiza un estudio previo de las diferentes técnicas utilizadas. En [52] se encuentra un estado del arte completo de los diferentes algoritmos de estimación del estado de carga para baterías plomo-ácido, siendo el mismo tipo de baterías usadas.

$$SoC(t) = SoC(t_0) - \frac{\eta}{C} \int_{t_0}^t i_{bat} \cdot dt \quad (3.18)$$

Donde:

SoC(t): Estado de carga en el instante t.

SoC(t<sub>0</sub>): Estado de carga "inicial".

η: Rendimiento de la batería.

i<sub>bat</sub>: Corriente de la batería.

C: Capacidad de la batería.

Se opta por usar el método de la integración de corriente o contador de Coulombios (ecuación 3.18), fusionado con la determinación del SoC a partir de la tensión en bornes de la batería y la temperatura ambiente, según el propio fabricante [14]. Así, el algoritmo finalmente usado se describe a continuación:

1. Se inicializa una variable para determinar si la batería está "fría" y así poder determinar su SoC solamente con la lectura de la tensión y temperatura ambiente. Inicialmente se presupone que la batería no ha sido usada y por tanto está "fría". En el momento que la corriente de la batería sea distinta a 0, la variable indicará que la batería está "caliente". Cuando la corriente de la batería sea nula, un contador se activará para calcular el tiempo que ha transcurrido sin corriente por la batería. Si este tiempo excede las 24 horas se puede determinar (según [14]) que la batería vuelve a estar "fría".
2. Si la batería está fría, se determina el SoC con su tensión siguiendo la tabla de la figura 3.29 y la compensación por temperatura en [14]. Si está "caliente" se determina el incremento de energía (desde el periodo anterior) como la integral de la corriente (ecuación 3.18) de la medida actual de corriente desde el periodo anterior (suponiendo la corriente constante entre periodos). Para una mayor precisión se mide exactamente el tiempo entre periodos, ya que ROS no se ejecuta en tiempo real.
3. Finalmente el incremento de energía es sumado al total acumulado, actualizándose la energía total de las baterías y guardándose para el siguiente ciclo. Esta energía es traducida a porcentaje del SoC para una mejor comprensión.

Voltage	State of charge
2.14Vpc	100%
2.10Vpc	80%
2.07Vpc	60%
2.04Vpc	40%
2.00Vpc	20%

Figura 3.29: Estimación del SoC a partir de la tensión en circuito abierto.

#### Alertas por valores límite

Tras disponer de todas las mediciones se determina si algún parámetro excede unos ciertos límites considerados como peligrosos para la integridad de la instalación.

#### Publicación en ROS

Finalmente, en la función principal de este nodo se publican en dos tópicos de ROS la información obtenida y procesada. Uno para todos los parámetros referidos a las medidas y otro para los valores de las alertas.

#### Iteración con "modo lectura" del HMI

Como funcionalidad extra, este nodo detiene la lectura de los sensores mediante comunicación I2C si se está realizando la misma lectura desde el HMI. Esta funcionalidad se implementa ya que si varios procesos actuando como "master" de un bus I2C intentan leer de un mismo sensor, la correcta obtención de la información se ve comprometida. Mediante la subscripción de un tópico de ROS proveniente del HMI se conoce si éste está realizando comunicación I2C con los sensores.

Cuando el HMI indica que está realizando dicha lectura, se detienen las comunicaciones I2C y se comienza a leer el SoC estimado por el HMI (también publicado en ROS). Así, una vez que se retomen las medidas desde el nodo, se sabrá el SoC actual sin tener que reiniciar el sistema.

#### ***Lectura Modbus***

Modbus es un protocolo de comunicaciones diseñado por Modicon en 1979 para su propia gama de controladores lógicos programables (PLCs). Está basado en la arquitectura maestro/esclavo (RTU)

o cliente/servidor (TCP/IP). Se ha convertido en un protocolo de comunicaciones estándar de uso en el ámbito de la industria, por lo que hay una mayor disponibilidad para la conexión de dispositivos electrónicos industriales [53]. Las principales razones por las que el Modbus se ha impuesto en el ámbito industrial por encima de otros protocolos son que se diseñó expresamente para aplicaciones industriales, el estándar es público y es fácil de implementar.

En este trabajo se recoge información del regulador DC que transforma la energía de los módulos fotovoltaicos. Al igual que en el nodo "Lectura de sensores", el nodo para la lectura Modbus se basa en una función principal asociada a un temporizador que la ejecuta cada tres segundos. Cada vez que se ejecuta la función, se inicializa la comunicación Modbus y se comienzan a leer los distintos registros uno por uno. Al finalizar la lectura, se publican los datos en un tópico de ROS y se cierra la comunicación Modbus.

Para la conexión del regulador DC y la Raspberry Pi 4 se ha usado un adaptador de puerto serie a Modbus para un conector tipo J12, para así poder conectar la Raspberry Pi 4 desde un puerto USB al regulador DC (figura 3.30). Para realizar bien la conexión ha de tenerse en cuenta las especificaciones del fabricante (figura 3.31).

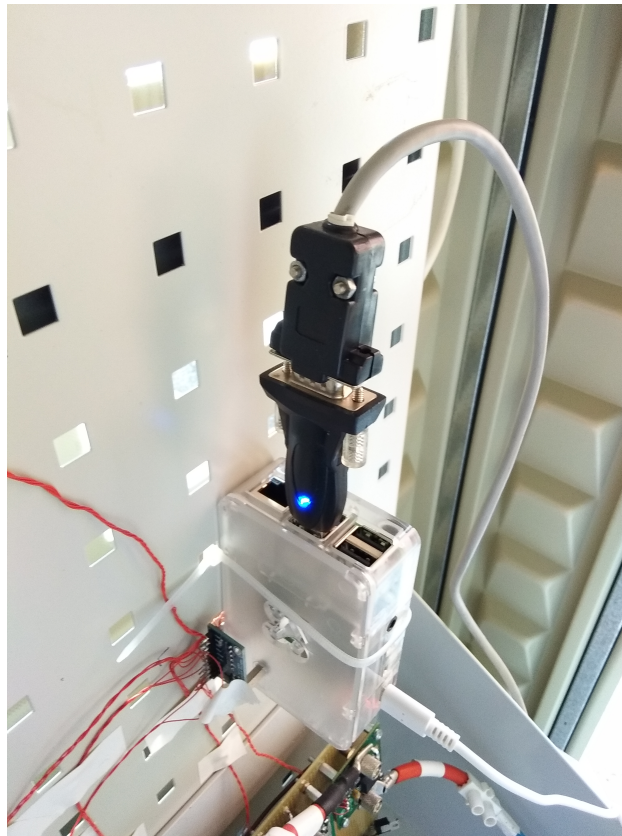


Figura 3.30: Conexión Modbus: Raspberry Pi 4.

### *Puente MQTT*

Este nodo es un puente entre las comunicaciones MQTT y los tópicos de ROS. Se crea ya que la estación meteorológica y el HMI publican sus datos a través de MQTT, en vez de directamente en ROS.

La estación meteorológica dispone de su propio microprocesador y dada su simplicidad (en comparación con la Raspberry Pi 4 y los procesos que alberga) se decide no incluir ROS en él.

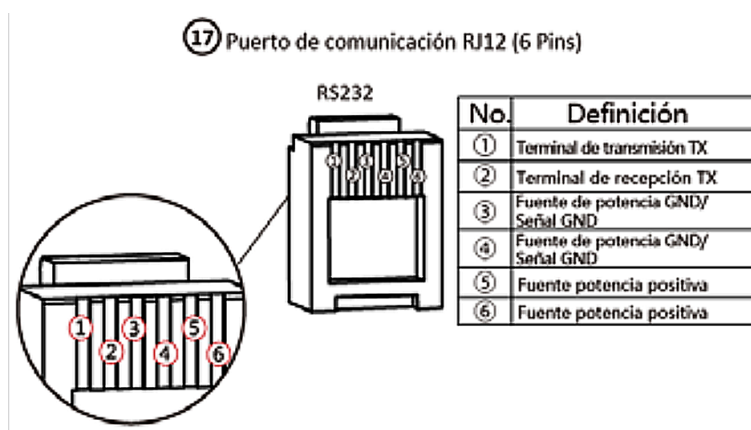


Figura 3.31: Puerto de comunicación RS232 - J12 (6 pines) [13].

Por otro lado, se usa MQTT en el HMI porque las funciones correspondientes a la publicación de tópicos en ROS (por parte de Node-RED) no funcionan como se esperaría y malogra el funcionamiento total del HMI.

La información que se envía a través de MQTT al Controlador DER es recogida y publicada en tópicos de ROS preparados de antemano. Esto quiere decir que el nodo no funciona para cualquier tópico MQTT, si no que previamente se han de conocer los tópicos MQTT y preparar el nodo en consecuencia. Así, el nodo se encuentra constantemente "escuchando" los tópicos MQTT esperados y cuando detecta nuevas publicaciones ejecuta una función *callback* para transferir la información a *topics* de ROS.

Para la realización de este nodo se ha hecho uso de la librería "Mosquitto" y del *broker* Mosquitto.

### 3.3.1.2. Nivel 2 - Lógica

El segundo nivel está referido a toda la parte de lógica y control de la microrred. Se compone principalmente del Servidor IEC61850 que a su vez contiene al EMS, principal responsable de los distintos comportamientos de la microrred.

#### ***Servidor IEC61850***

El Servidor del estándar IEC61850 es el encargado de controlar la actuación de la microrred, de actualizar el modelo IEC61850 de la planta (la microrred) y de realizar las comunicaciones de alto nivel con otras entidades (Clientes IEC61850).

#### Modelo IEC61850 de la planta

La construcción del Servidor comienza por el diseño del modelo. Aunque el diseño de éste queda fuera de este trabajo, se añade como anexo ya que forma parte del sistema completo de la microrred y se hace uso de él en este trabajo.

El diseño del modelo IEC61850 de la planta de la microrred (anexo C) ha sido llevada a cabo por dos compañeros del grupo de investigación GEISER de la Universidad de Alcalá.

#### Integración del estándar IEC61850

El nodo del Servidor, al igual que los nodos anteriores, se integra dentro de ROS. Este cuenta con una estructura similar al resto, pero a excepción del resto, debe contener unas librerías específicas del



estándar IEC61850 creadas en C [54]. Se usan estas y no otras porque son públicas y mantienen la filosofía de este trabajo.

Dado que el compilador por defecto de ROS es para C++, se encuentran muchos problemas a la hora de compilar el código C de la librería IEC61850, concretamente funciones de comunicación y modificación de datos, y tipos de datos específicos del estándar. Es por ello que se decide compilar la librería de forma externa e incluirla dentro de ROS a mano. Para ello se compila la librería según sus propias instrucciones. La librería IEC61850 posee por defecto los archivos necesarios para ser compilada a través del comando `make`.

Tras esto, en el archivo `"CMakeLists.txt"` correspondiente al paquete de ROS que contiene el nodo del Servidor, se incluyen los archivos *object* (`.o`) necesarios para la creación del archivo ejecutable del nodo.

La librería IEC61850 cuenta con funciones específicas para las comunicaciones de alto nivel y modificación de datos del estándar. Un ejemplo de estas es `IedServer_updateDbposValue(iedServer, stVal, control)` cuya función es la de actualizar una variable DBpos de tipo `stVal` en el modelo IEC61850 asociado al identificador `iedServer` con el valor de `control`. Otra función ejemplo es `IedServer_lockDataModel(iedServer)` que sirve para bloquear el mutex asociado por defecto al modelo IEC61850 asociado al identificador `iedServer` y evitar conflictos en los accesos a memoria. De igual forma, para desbloquear el mutex se usa la función `IedServer_unlockDataModel(iedServer)`.

Finalmente se ha de indicar que el Servidor IEC61850 requiere ser ejecutado como superusuario (`sudo` en Linux), pero los nodos en ROS no admiten el comando `sudo`. Por ello, una vez generado el ejecutable, cambiamos sus permisos por terminal (el ejecutable en este caso se llama `ems_node`) para que pueda ejecutarse como superusuario por defecto:

```
1 sudo chown root:root ems_node
2 sudo chmod a+rx ems_node
3 sudo chmod u+s ems_node
```

Tras la modificación de los permisos, ya puede ser ejecutado con normalidad: `./ems_node`.

### Estructura del nodo

En primer lugar se declaran todas las funciones a usar del estándar IEC61850 y se inicia el Servidor IEC61850 (en la función "main") siguiendo la siguiente secuencia:

1. Reservar un puerto TCP para el Servidor.
2. Crear un nuevo objeto de configuración de servidor.
3. Establecer la configuración del servidor.
4. Configuración de los permisos de operación sobre los dispositivos del modelo.
5. Asociar interrupciones a las funciones de operación sobre el modelo.
6. Establecer los permisos de operación desde el cliente.
7. Iniciar el servicio MMS para escuchar conexiones de clientes.

Tras la inicialización, el proceso del Servidor recoge constantemente toda la información de los distintos parámetros de la microrred a través de tópicos de ROS. Tras su obtención modifica las variables



correspondientes del modelo de la instalación según el estándar IEC61850. Si algún parámetro se encuentra fuera del rango de funcionamiento normal (*topic* ROS de alertas) se informa a la máquina de estados encargada del control (EMS), modificando así su comportamiento.

Al mismo tiempo atiende las peticiones de Clientes IEC61850, que pueden pedir información del modelo o pueden requerir de actuar sobre la planta modificando el modelo IEC61850. En este último caso, el Servidor comprueba primero que el Cliente tenga permiso para controlar la microrred y seguidamente queda registrado en el propio modelo que un Cliente remoto está gobernando la planta. En la figura 3.32 se muestra la secuenciación lógica del nodo.

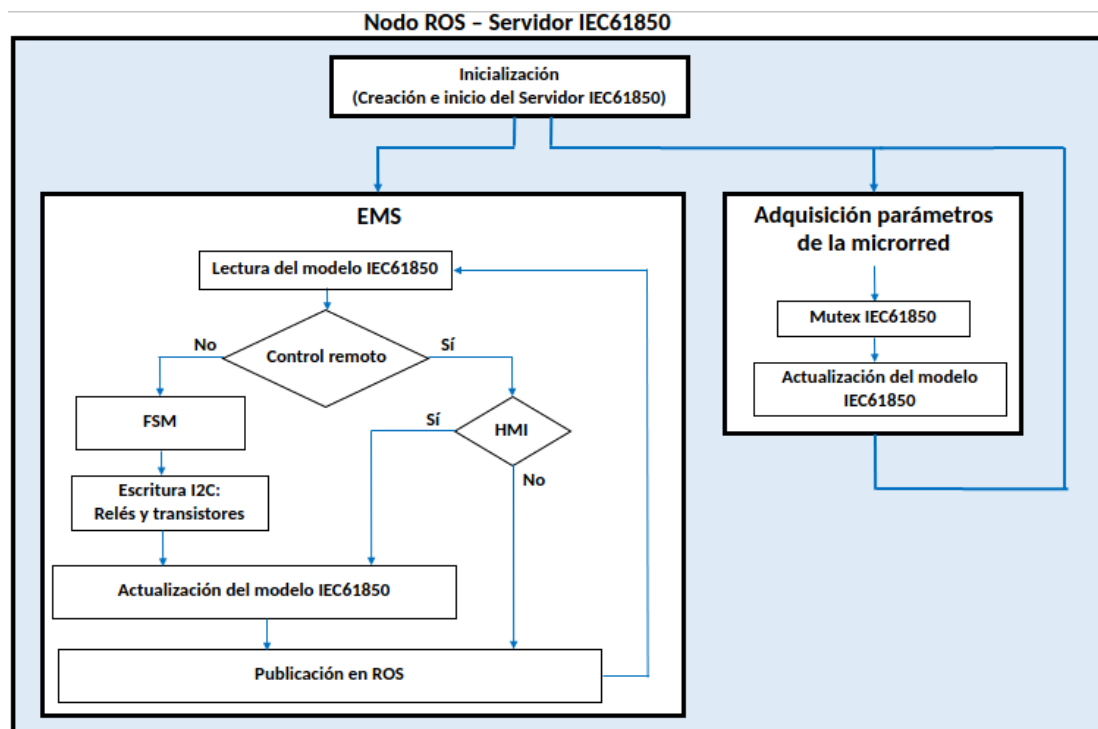


Figura 3.32: Esquema conceptual del nodo ROS Servidor IEC61850.

### EMS

En este TFM se implementa un EMS a modo demostrativo, siendo su justificación objeto de estudio de otro trabajo complementario. Este EMS se ha integrado dentro del Servidor IEC61850 porque sus funcionalidades están estrechamente relacionadas con las del Servidor y de este modo se simplificaba su implementación.

Las funciones del EMS son las del control de los flujos de energía en la microrred. La parte principal del EMS es la máquina de estados (FSM) que controla las transiciones entre diferentes comportamientos de la planta. El EMS realiza también tareas de escritura I2C para controlar la apertura y cierre de los distintos relés y transistores de la instalación.

### Estructura del EMS

El EMS se ejecuta de forma periódica en una función *callback* de ROS asociada a un temporizador. Para las pruebas realizadas se establece un periodo de 5 segundos.

La primera vez que se ejecuta el EMS se inicializan todas las variables correspondientes con los relés y transistores en el modelo IEC61850. Consecuentemente se escribe en el expansor I2C esta configuración

inicial para llevar a la microrred a su estado inicial. Tras esto se espera durante 1 minuto para obtener la primera medida de potencia fotovoltaica generada, ya que se conectan solamente las tres cargas controlables al bus DC (tabla 3.2 y figura 3.33).

Después de la inicialización se procede con el funcionamiento normal del EMS:

1. Se leen los parámetros del modelo IEC61850, incluidos los referidos a relés y transistores por si hubiesen sido modificados externamente.
2. Se obtiene la hora del día, ya que también es determinante en el funcionamiento de la FSM.
3. Se comprueba si algún cliente a iniciado un control remoto de la microrred.
4. Se comprueba (de forma indirecta) si la red AC se ha caído del lado del equipo de fuerza. Aquí ya se entra en la FSM, pues una caída de la red provocaría un cambio inmediato de estado.
5. Un contador calcula un periodo 10 minutos para actuar en un funcionamiento normal sobre el cambio de estados. Durante ese periodo se calcula el promedio de la potencia fotovoltaica generada. Se hace así para no estar permanentemente conmutando de estados ante posibles cambios transitorios de la potencia fotovoltaica (por ejemplo el paso de nubes). Además, durante el último minuto se observa la mínima potencia fotovoltaica generada ya que es una información útil para la FSM.
6. Tras el periodo de 10 minutos se ejecuta la parte principal de la FSM. Su funcionamiento viene descrito en la tabla 3.2 y figura 3.33.
7. Se traduce el estado actual de la FSM a actuaciones en los relés y los transistores de la microrred.
8. Se actualiza el modelo IEC61850 con la nueva disposición de los elementos de conmutación.
9. Finalmente se publica en ROS toda la información relacionada con la FSM y los distintos relés y transistores.

Estado	Relé 1	Relé 2	Batería	Cargas controlables
0	OFF	OFF	OFF	OFF
1	ON	OFF	ON	OFF
2	OFF	ON	OFF	OFF
3	ON	ON	ON	OFF
4	OFF	OFF	ON	ON (3kW)
5	ON	OFF	ON	ON (3kW)
6	OFF	ON	OFF	ON (3kW)
7	ON	ON	ON	ON (3kW)

Tabla 3.2: Estados de la máquina de estados (FSM) del EMS.

#### Interacción con "modo operativo"

El nodo del Servidor se suscribe a un tópico de ROS proveniente del HMI, donde se indica si éste a iniciado un control manual de la microrred. En ese caso, la FSM del EMS detiene su funcionamiento y se bloquea la actuación sobre los elementos de la planta (relés y transistores).

La información de estos elementos de conmutación durante el "modo operativo" es publicada en ROS desde el HMI. El nodo del Servidor recoge esta información y mantiene actualizando el modelo IEC61850 en todo momento.

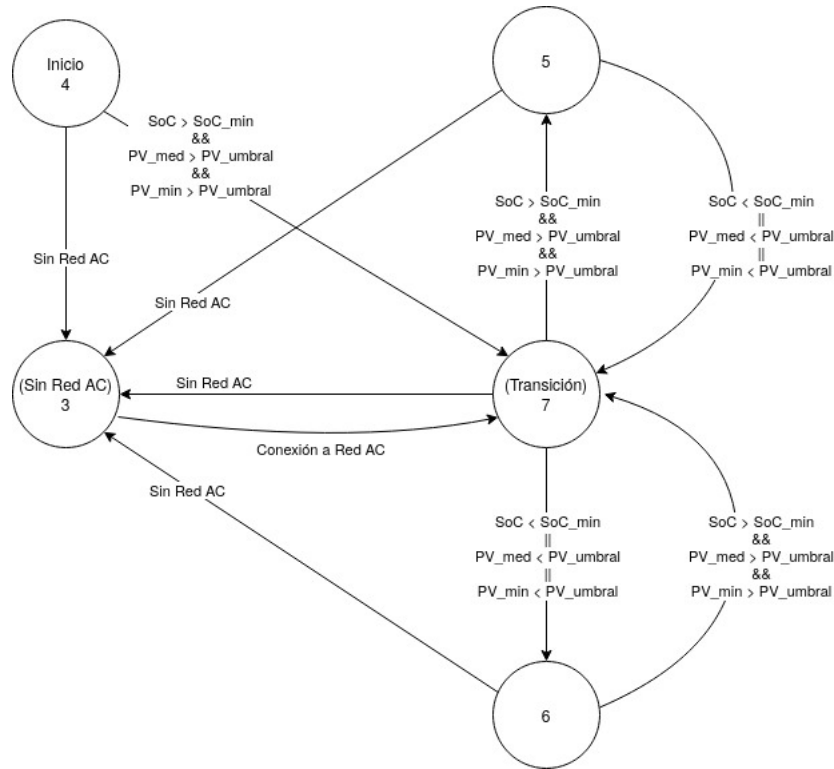


Figura 3.33: Transiciones de la máquina de estados (FSM) del EMS.

### 3.3.2. HMI (Nivel 3)

Con el propósito de mostrar todos los datos recogidos de la planta, mostrar las distintas variables de control y la posibilidad de actuar sobre la planta de forma manual, se implementa una interfaz gráfica. Esta interfaz estará en continua comunicación con el Controlador DER para coordinar la actuación y la medición de la planta.

La interfaz gráfica cuenta con dos modos de funcionamiento principales:

- *Modo operativo.* Este modo sirve para la realización de pruebas, controlando todos los parámetros de forma manual y pudiendo tomar las medidas de forma externa al Controlador DER. Además se recomienda que en la puesta en marcha de la instalación se utilice el modo operativo para una revisión inicial de los distintos elementos.
- *Modo normal.* Este modo actúa simplemente como una interfaz gráfica de las variables y medidas manejadas por el Controlador DER. Además se implementan una serie de funcionalidades para pasar del modo normal al modo operativo (y viceversa) sin la necesidad de reiniciar todo el sistema.

Para la implementación de la interfaz gráfica se utiliza Node-RED (logotipo mostrado en la figura 3.34). Esta es una herramienta de desarrollo software para interconectar distintos dispositivos hardware, APIs y servicios online de forma visual y así más intuitiva. Cuenta con una amplia variedad de librerías y una gran comunidad que mantiene su buen funcionamiento ya que es software libre [55].

Se elige Node-RED por ser software libre y seguir así la filosofía de todo este trabajo, además de la gran cantidad de librerías y documentación disponible. También el hecho de que toda su interfaz, incluido su *dashboard*, esté basado en navegador *web* facilita su manejo desde equipos remotos siempre que estén dentro de la misma red.

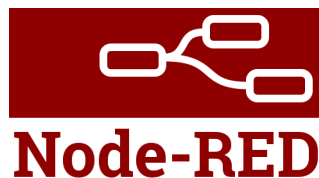


Figura 3.34: Logotipo de Node-RED.

Se valoran otras opciones como utilizar LabView, pero por problemas de incompatibilidades con Raspberry Pi OS (Raspbian) se descarta. También se valora MyOpenLab, pero se descarta ya que su *dashboard* no está basado de forma nativa en navegador *web*. En la figura 3.35 se muestra un fragmento de la programación en el entorno de Node-RED.

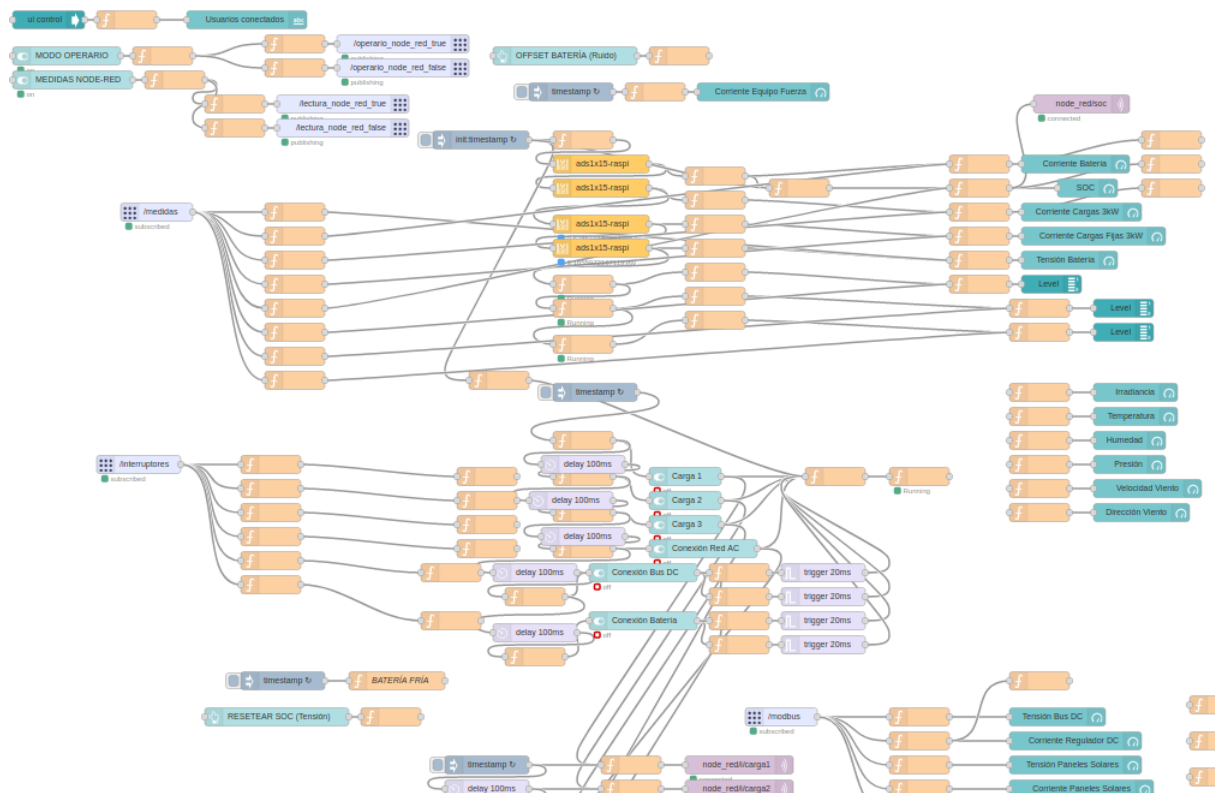


Figura 3.35: Programación por bloques en Node-RED.

### 3.3.2.1. Modo operario

El modo operario sirve para modificar manualmente todos los parámetros de control de la planta, con la posibilidad de tomar medidas independientemente de procesos externos. Este modo cuenta con las siguientes funcionalidades:

- *Lectura de sensores (I2C)*

La comunicación I2C con el dispositivo ADS1115 se realiza gracias a unos bloques especialmente dedicados a este dispositivo, y la comunicación con el sensor BME680 se realiza mediante bloques de función programados en Python. La petición de información a ambos se realiza con un periodo de 2 segundos y de forma secuencial. Adicionalmente se añade un botón para establecer manualmente el offset de la medida de la corriente de la batería, al valor actual.

Las medidas son representadas en la página principal del *dashboard* (figura 3.36), al igual que el botón de offset.

- *Gestión del SoC*

El SoC es calculado siguiendo un algoritmo desarrollado en la sección 3.3.1.1 ("*Nivel 1 - Interacción física: Lectura Sensores*"). Se habilita un botón para poder reiniciar el contador del que hace uso el algoritmo para determinar si el SoC puede ser determinado directamente con su tensión.

Después de ser calculado, el valor del SoC es publicado en ROS (a través de protocolo MQTT) para que al pasar al modo normal, se pueda retomar el funcionamiento del EMS del Controlador DER sin reiniciar el sistema. El SoC también es representado en la página principal del *dashboard*.

- *Lectura del regulador DC (Modbus)*

La lectura de los parámetros del regulador DC es llevada a cabo (en un principio) por unos bloques específicos de comunicación Modbus. Los parámetros leídos son la tensión del conjunto de módulos fotovoltaicos, su corriente, la potencia generada, y la tensión y corriente a la salida del regulador DC. Posteriormente, se vio que era incompatible con el nodo de Modbus del controlador DER porque no se cerraba el puerto y bloqueaba la comunicación.

La tensión y corriente a la salida del regulador son representadas en la página principal del *dashboard*, y los parámetros de los módulos fotovoltaicos en una página secundaria. Las direcciones de los registros Modbus del regulador son expuestas con más detalle en la sección 3.3.1.1 ("*Nivel 1 - Interacción física*").

- *Control de interruptores*

El control de la apertura y cierre de las cargas regulables y los relés se lleva a cabo desde el *dashboard* mediante unos interruptores (figura 3.36). La información de los interruptores es codificada y enviada por I2C (mediante un bloque de función programado en Python) al expansor I2C.

La información del estado de los diferentes interruptores es publicada en ROS para que al pasar al modo normal, se pueda retomar el funcionamiento del EMS del Controlador DER sin reiniciar el sistema.

- *Implementación de contraseña*

Se implementa contraseña para proteger la entrada al entorno gráfico de programación y al *dashboard* (entorno de monitorización y control). Para habilitar la protección con usuario y contraseña hay que modificar el archivo de configuración settings.json.

- *Visualización de usuarios activos*

Finalmente se añade una funcionalidad con la que cualquiera que acceda al *dashboard* pueda ver cuántos usuarios hay conectados en ese momento. Esto es muy útil si se quiere hacer pruebas, para no entrar en conflicto con otro operario.

### 3.3.2.2. Modo normal

El modo normal está referido a la función de Node-RED como pura interfaz gráfica, recogiendo los datos del Controlador DER y representándolos en el *dashboard web*. Este modo cuenta con las siguientes funcionalidades:

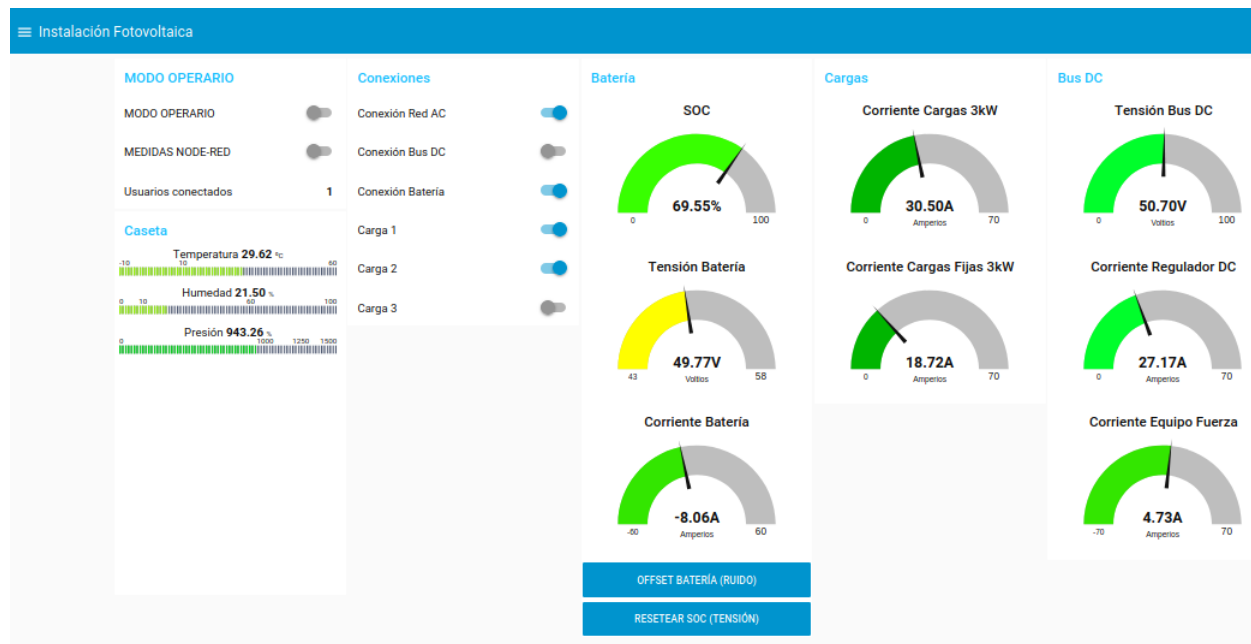


Figura 3.36: Dashboard principal de Node-RED.

#### ■ *Recogida de medidas*

Las medidas son recogidas a través de ROS (ya que son publicadas en dicha plataforma), subscribiéndose a los *topics* de medidas de sensores y de parámetros del regulador DC. Esta información es mostrada en el *dashboard* de igual modo que se detallaba en el apartado anterior.

Se añade también un interruptor para conmutar entre si las medidas son recogidas del Servidor o si son realizadas por el propio Node-RED (lo que sería una parte del modo operativo, y podría denominarse como "modo lectura"). Esto se implanta porque si varios procesos actuando como "master" de un bus I2C intentan leer de un mismo sensor, la obtención de la información falla. Mediante la publicación de un *topic* en ROS se comunica al Servidor/EMS si se ha activado este modo de lectura y así detendría la comunicación I2C y se recogería las medidas desde Node-RED (publicadas en ROS como se exponía en apartado anterior).

#### ■ *Recogida de estado de interruptores*

De igual modo que en la recogida de medidas, los estados de los interruptores son recogidos de un *topic* de ROS y representados en el *dashboard*.

También se añade un interruptor para conmutar entre el modo operativo y el modo normal, publicando un *topic* en ROS con esta información para que el EMS sepa que no tiene que realizar operaciones de escritura en el expansor I2C. Además, en modo normal, si de forma manual se acciona algún interruptor y no se ha activado el modo operativo, la acción queda bloqueada sin entrar en conflicto.

# Capítulo 4

## Resultados

El driver hardware, el controlador DER y la estación meteorológica han sido probado y validado a lo largo de todo su desarrollo y se han incorporando de forma paulatina a la instalación de la microrred existente. Siempre, antes de incorporar cada elemento a la instalación, se probaba su funcionamiento en el laboratorio.

En la primera parte de este capítulo se muestra el resultado de la parte hardware de toda la instalación de la microrred y de los distintos elementos desarrollados. En la segunda parte se muestran los datos obtenidos que validan el funcionamiento global de todo el sistema durante días enteros. La microrred se prueba en días con condiciones atmosféricas diversas (soleados y nublados), comprobando el funcionamiento de la máquina de estados del EMS y sus transiciones ante cambios de irradiancia.

### 4.1. Hardware

La primera figura (figura 4.1) muestra la versión final de la placa principal del driver hardware. Se encuentra anclada a un panel metálico colocado sobre la estructura del equipo de fuerza. Tanto la pintura exterior del equipo de fuerza como la pintura del panel metálico son aislantes, proporcionando el aislamiento necesario entre ambos elementos (necesario ya que el chasis del equipo de fuerza está llevado a tierra por razones de seguridad). Igualmente se ha evitado cualquier contacto metálico entre la placa principal y el panel metálico.

En la figura 4.2a se muestra la SBC (Raspberry Pi 4) junto con el ADC y el sensor ambiental BME680. Como se exponía en el capítulo anterior, estos elementos se encuentran separados de la placa principal pero formando parte del driver hardware. Se encuentran en el mismo panel metálico que la placa principal, en la parte superior. Se evitan también los contactos metálicos con el panel, para mayor seguridad.

En la parte superior de la figura 4.2b se muestra un diodo de protección que fue retirado en la versión final de la instalación. Este diodo bloqueaba corrientes de entrada al regulador DC, ya que en pruebas iniciales se produjo un retorno forzado de corriente por él, debido a que no se habían instalado aún las cargas de 1kW. Tras las pruebas iniciales se retiró pues ya no era necesario.

En la parte inferior de la figura 4.2b se observa el montaje de las conexiones del bus DC. Dado que el bus tiene un número considerable de conexiones y todas ellas de corriente elevada, se decide montar una pequeña caja de conexiones anclada al panel metálico. De esta forma se pueden asegurar correctamente tanto los cables conectados a la parte positiva del bus y los dirigidos a la parte negativa de este.



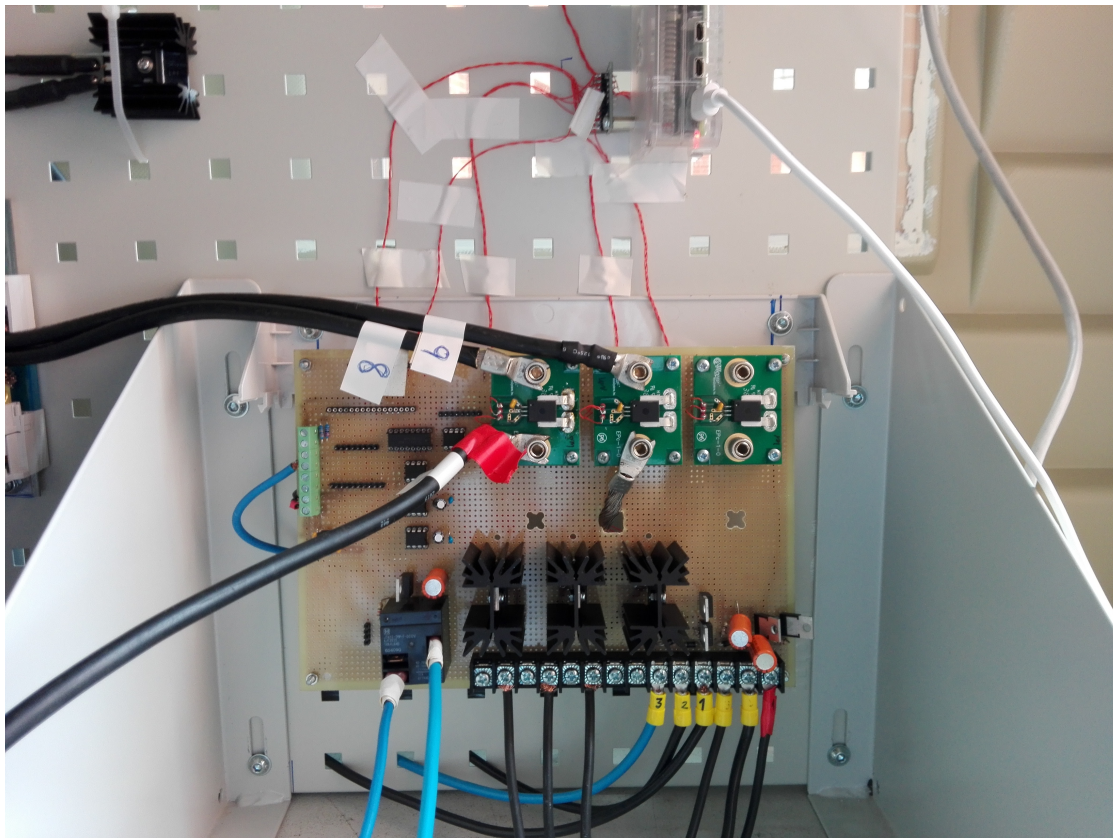
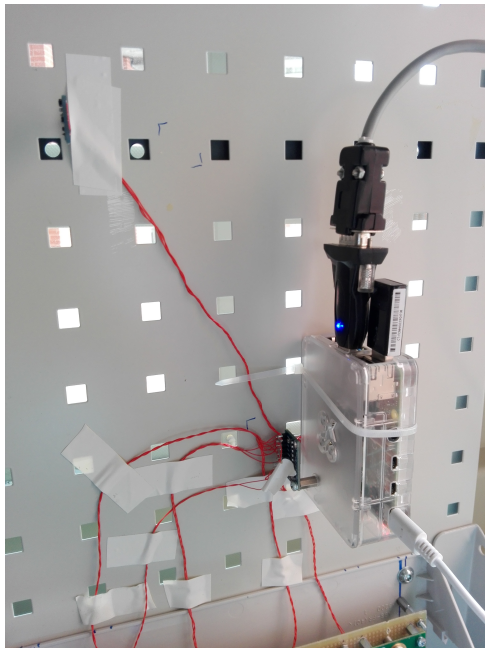
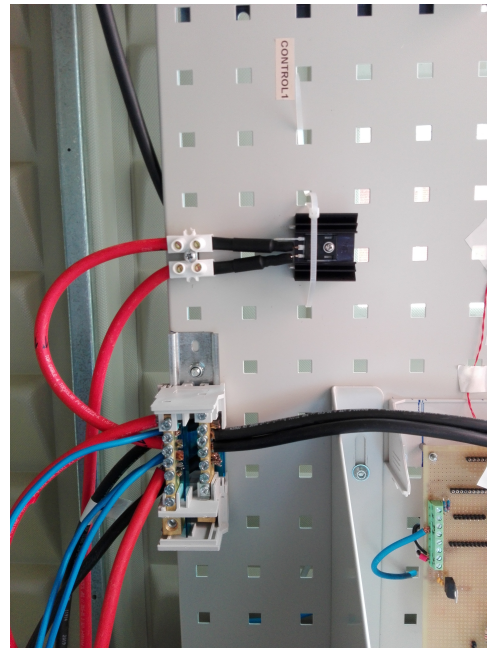


Figura 4.1: Driver hardware - Placa principal.



(a) Driver hardware: SBC, ADC y BME680.



(b) Montaje de nodos Bus DC y diodo de protección.

Figura 4.2: Elementos en el panel metálico no pertenecientes a la placa principal.



Finalmente, en la figura 4.3 se muestra el conjunto de la instalación alojada en el interior de la caseta. Arriba a la derecha se encuentra el panel metálico con los elementos descritos anteriormente. Debajo del panel y sustentándolo, se encuentra el equipo de fuerza que actúa como rectificador. Abajo al fondo se encuentran las baterías (de color amarillo). Y a la izquierda, en otro panel metálico, se encuentra el regulador DC junto con tres interruptores, cada uno de ellos asociado a una fila de los módulos fotovoltaicos, pudiendo variar la potencia conectada al regulador DC.

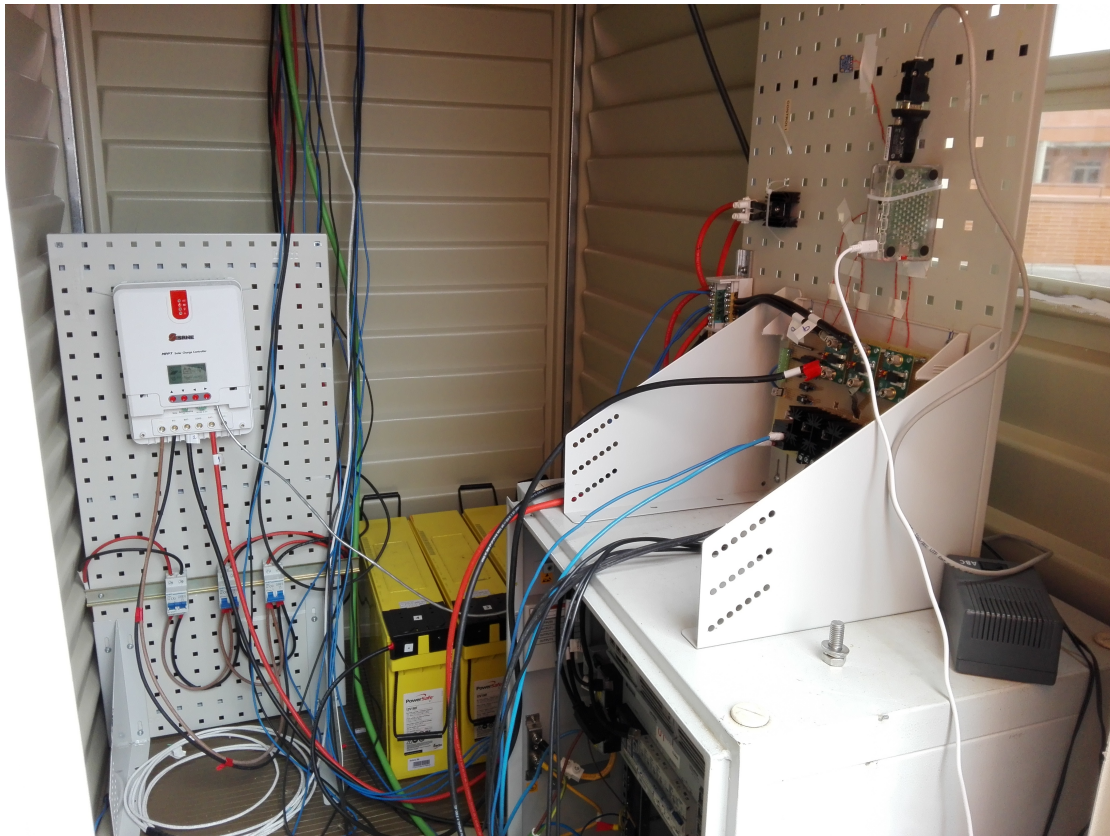


Figura 4.3: Conjunto de la instalación (interior de la caseta).

Por otra parte, separado de la caseta, se encuentra la instalación de los módulos fotovoltaicos y la estación meteorológica. En las figuras 4.4a y 4.4b se muestra la instalación de estos elementos. Aunque el diseño e instalación de los módulos fotovoltaicos no forma parte del desarrollo de este trabajo se considera de interés mostrarlos ya que sí forman parte del conjunto de la microrred.

Por último, en la figura 4.5 se muestra la electrónica de acondicionamiento de la estación meteorológica, expuesta en el capítulo anterior. Esta electrónica se encuentra bajo la instalación de los módulos fotovoltaicos, al resguardo de la intemperie.



(a) Instalación módulos fotovoltaicos y pirómetro.



(b) Anemómetro y veleta de la estación meteorológica.

Figura 4.4: Instalación parte superior.

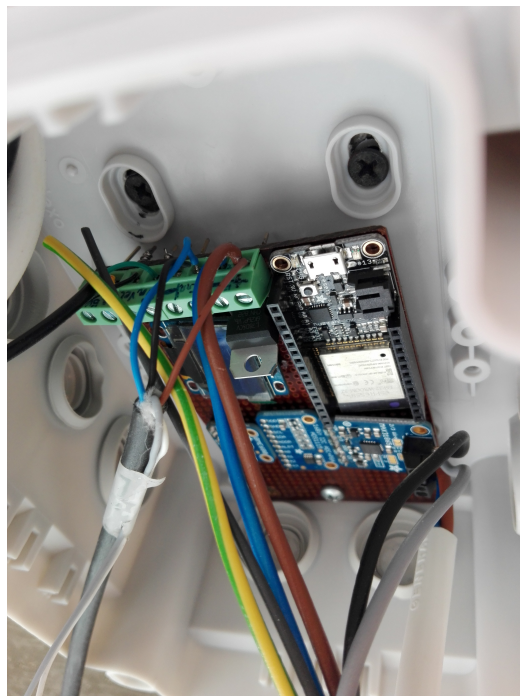


Figura 4.5: Estación meteorológica - Electrónica de adquisición.

## 4.2. Casos de uso

En esta sección se expone el funcionamiento de la microrred ante diferentes condiciones ambientales. Además, se comprueba el modo operario (o modo manual de control) y la adquisición de la información meteorológica en la microrred y en la nube IoT.

En todos los casos se han usado únicamente 2kW de cargas controlables y 1kW de cargas fijas. Esto es debido a que la conexión de la microrred a la red AC se realiza a través de una cometida de uso genérico, con un límite de potencia de alrededor de 3kW. Cuando la potencia fotovoltaica baja demasiado, gran parte de la energía que demandan las cargas es suministrada desde la red AC a través del equipo de fuerza. Por lo tanto, cuando se conectan más cargas (superando los 3kW) y baja la potencia fotovoltaica, se consumen más de 16A AC y saltan los interruptores magnetotérmicos, provocando la desconexión permanente (solo reversible manualmente) de la microrred de la red AC.

### 4.2.1. Día soleado - 8 de septiembre de 2020

En esta primera subsección se expone la información recogida del funcionamiento de la microrred en el transcurso de un día totalmente soleado, sin ninguna nube. En la figura 4.6 se comprueba como la gráfica de irradiancia aumenta suavemente durante el día hasta su máximo sobre las 14:00 y baja paulatinamente hasta el ocaso. En la misma figura se observa la potencia fotovoltaica producida, alcanzando un máximo de 2kW que corresponde con las dos cargas controlables asociadas a esta parte. Se comprueba que desde las 12:00 a las 17:00 aproximadamente, las cargas controlables son alimentadas plenamente con energía solar. El regulador DC es el encargado de mantener esta potencia máxima constante, ya que se configura como MPPT.

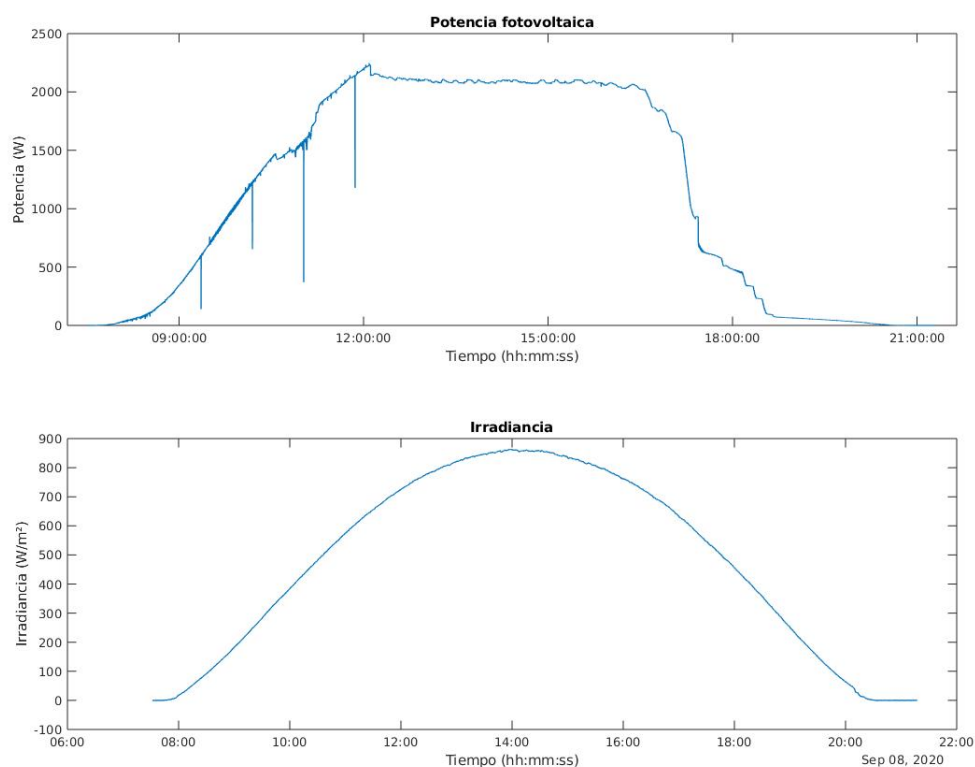


Figura 4.6: Potencia fotovoltaica vs Irradiancia.

En la figura 4.9 se comprueba que la FSM se encuentra en estado 5 durante el periodo de máxima potencia. Además, en la figura 4.7 se observa como durante este periodo la batería absorbe un excedente de energía, aumentando su SoC.

Durante los intervalos adyacentes al de máxima potencia, se comprueba como la batería ayuda a aportar la energía restante a las cargas controlables, ya que la FSM se encuentra en estado 5 pero aún la energía fotovoltaica no es suficiente. Se puede observar también en la figura 4.7 de la batería, como la tensión aumenta cuando absorbe energía y como disminuye cuando aporta energía. Del mismo modo, se puede comprobar como varía la corriente en los distintos intervalos (negativa cuando aporta corriente, y positiva cuando la absorbe).

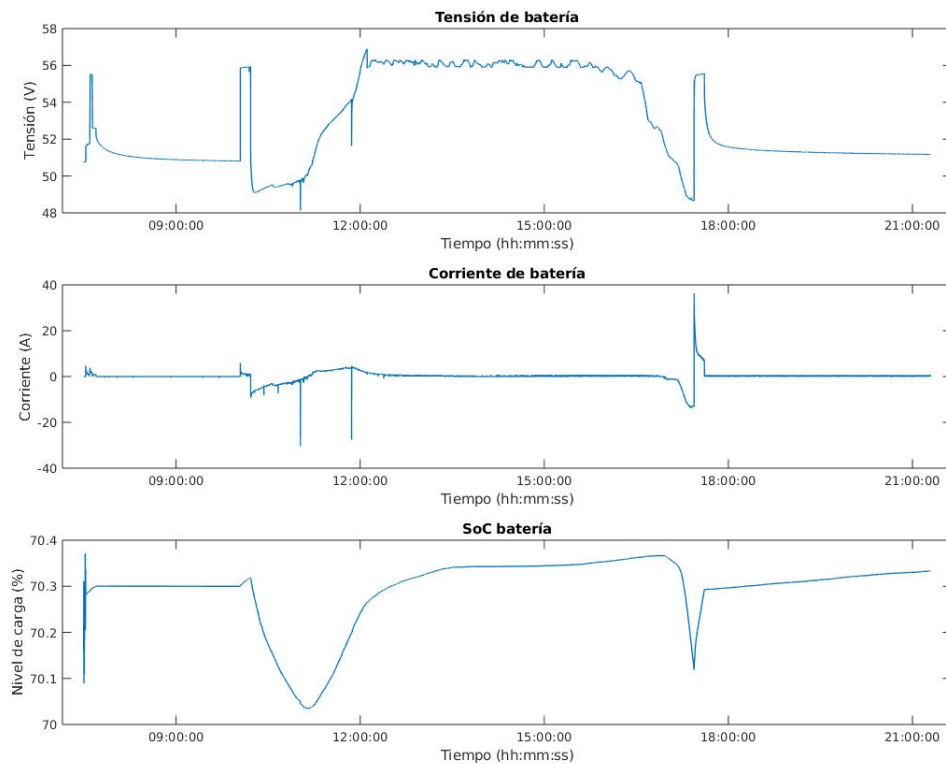


Figura 4.7: Parámetros de las baterías (BESS).

Por otra parte, en la figura 4.8 se recogen todas las corrientes de de la microrred, para tener una imagen global de estas. Destacar que la corriente de las cargas controlables varía en función del estado de la FSM del EMS y de la potencia fotovoltaica generada. También, la corriente que aporta el regulador DC sigue el mismo perfil que la potencia fotovoltaica generada, lo que implica que el funcionamiento en MPPT mantiene fija la tensión de los módulos fotovoltaicos (hecho que se comprueba en la figura 4.10 con los parámetros medidos por el regulador DC).

Se observa también en la figura 4.10 que la tensión del bus DC no sólo depende de la potencia fotovoltaica, si no que varía con la conexión de los distintos elementos de la microrred (cargas, batería y equipo de fuerza).

Finalmente, en la figura 4.9 se puede analizar el comportamiento del EMS, el promedio y el mínimo de potencia fotovoltaica generada (usado en la lógica de la FSM) y las distintas variables de conexión de los elementos de la microrred.



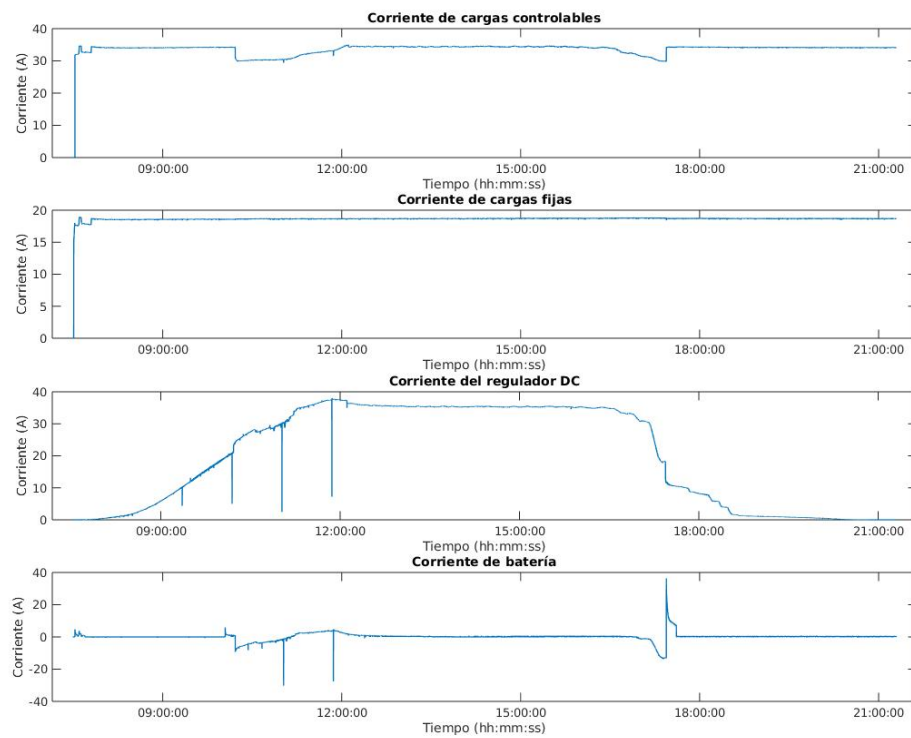


Figura 4.8: Corrientes de la microrred.

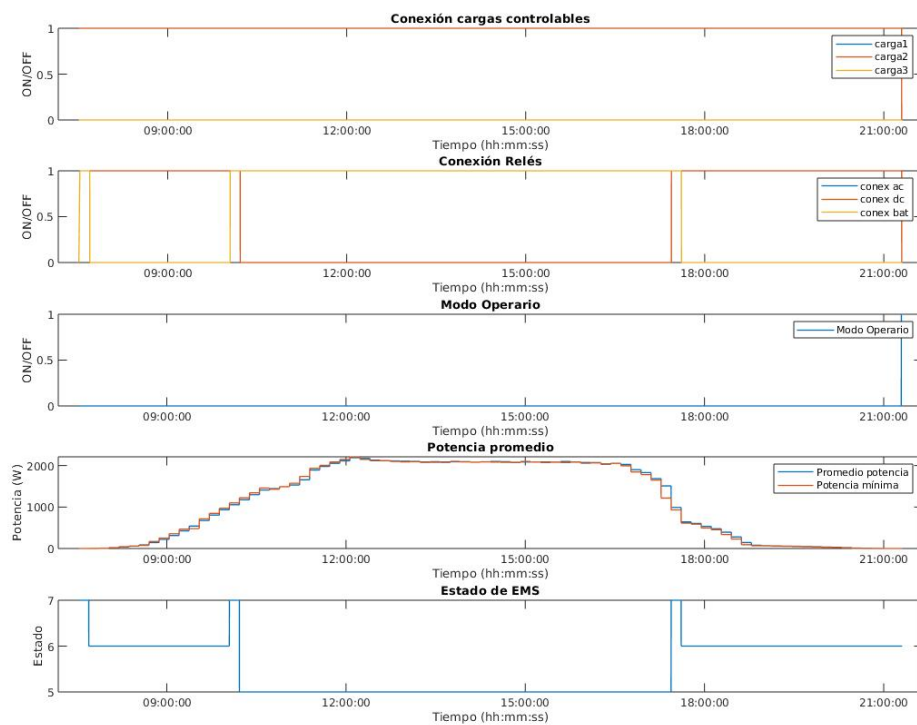


Figura 4.9: Estados y variables de control del EMS.

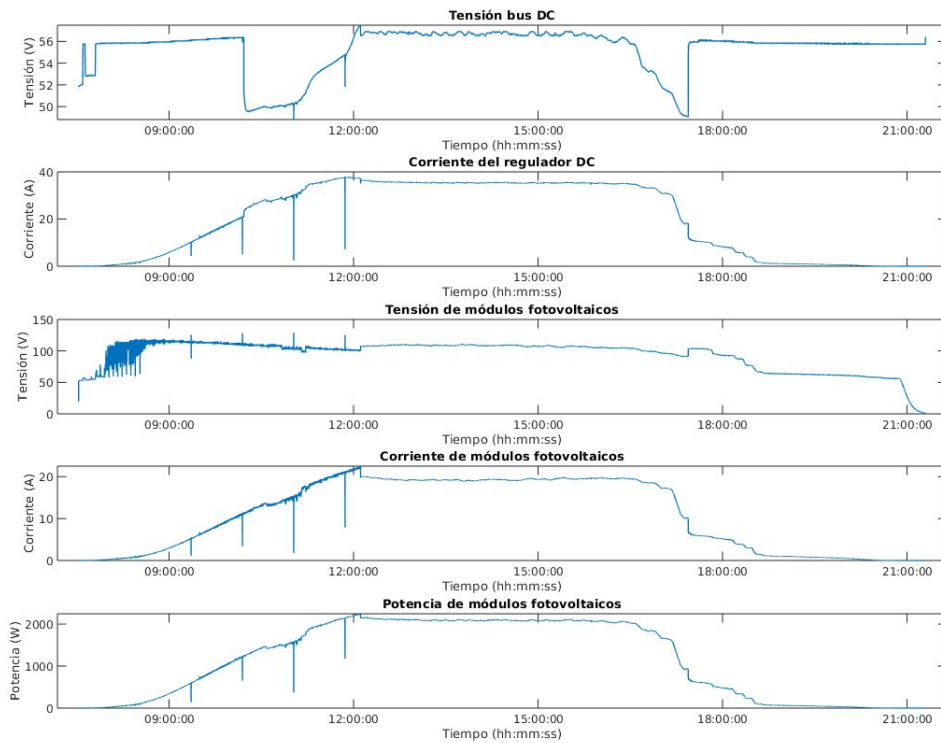


Figura 4.10: Parámetros de los módulos fotovoltaicos.

#### 4.2.2. Día nublado - 14 de septiembre de 2020

En esta subsección, al contrario que en la anterior, se exponen los datos del funcionamiento de la microrred recogidos durante un día totalmente nublado. En la figura 4.11 se observa perfectamente la fluctuación de la irradiancia, dependiendo de la nubosidad. En algunos momentos se alcanzan picos de  $1000 \text{ W/m}^2$ , lo que podría parecer un error de medida, pero esto es debido a que en días nublados cuando aparece un claro se suman la irradiancia directa del Sol y la reflejada en las nubes [56].

La potencia generada sigue el mismo perfil que el de la gráfica de la irradiancia, por lo que se comprueba que la potencia fotovoltaica generada sigue una relación lineal respecto a la irradiancia.

En este día nublado, la máquina de estados del EMS (figura 4.14) se encuentra gran parte del día en estado 6 (lo que significa que las cargas se alimentan mayoritariamente desde el equipo de fuerza), aunque desde aproximadamente las 12:00 a las 16:00 hay dos periodos en los que la FSM se encuentra en estado 5 (se supera el umbral de potencia fotovoltaica y se desconecta el equipo de fuerza del bus DC).

En las gráficas de la batería (figura 4.12) se puede observar lo comentado en el anterior párrafo: hasta pasadas las 12:00 no comienza a aportar energía al bus DC, viéndose como el SoC cae casi un 1 % hasta aproximadamente las 16:00. Las variaciones de tensión (subidas hasta casi los 56V) en el resto del tiempo son debidas al estado 7 de la FSM (estado transitorio entre el 5 y el 6) donde la batería es conectada al bus DC y absorbe un poco de energía.

Finalmente, en la figura 4.13 se observan varios picos de corriente positiva en la batería. Estos picos coinciden con los picos de generación de energía fotovoltaica comentados anteriormente, por lo que se

deduce que en esos momentos la batería absorbe el excedente de energía. Este hecho también se puede contrastar con los datos recogidos de los módulos fotovoltaicos (figura 4.15).

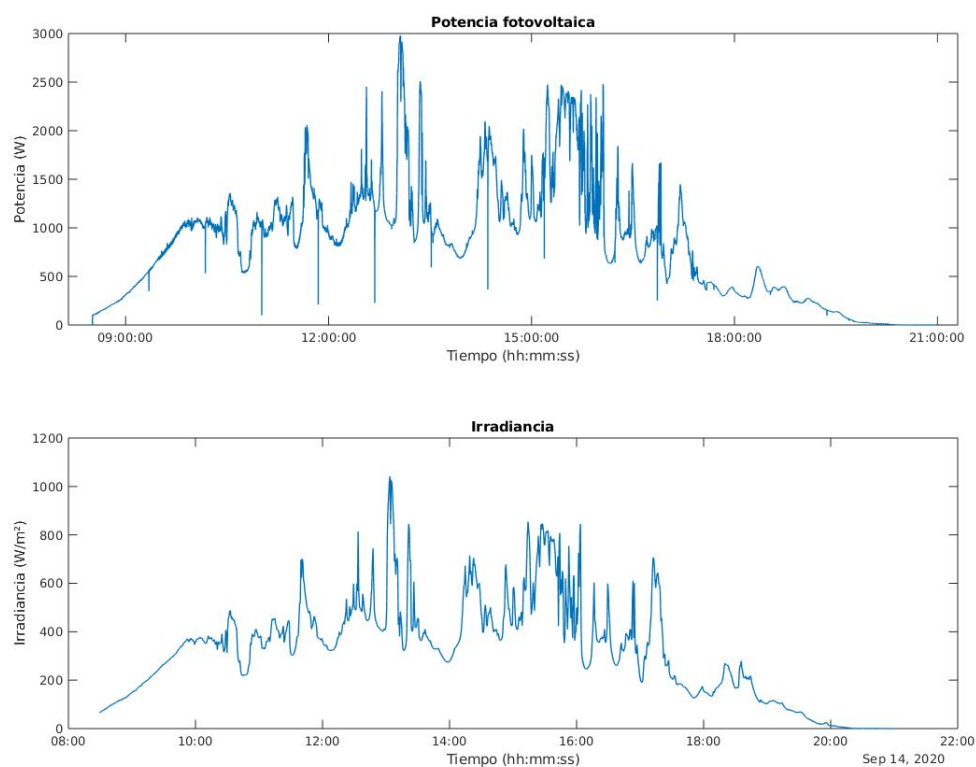


Figura 4.11: Potencia fotovoltaica vs Irradiancia.

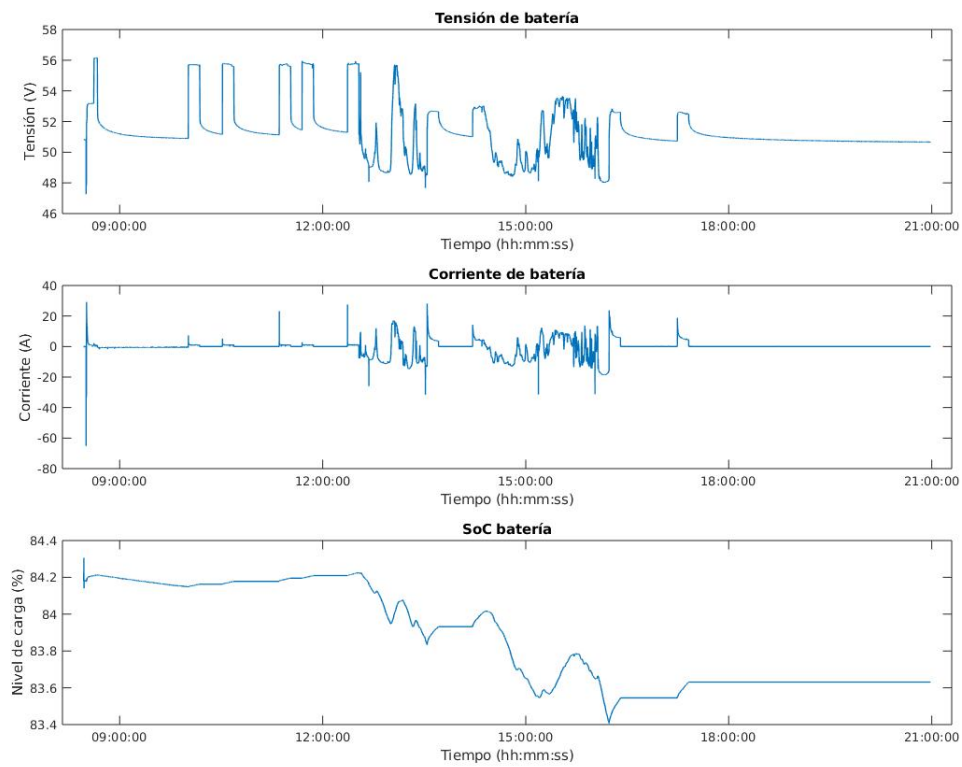


Figura 4.12: Parámetros de las baterías (BESS).

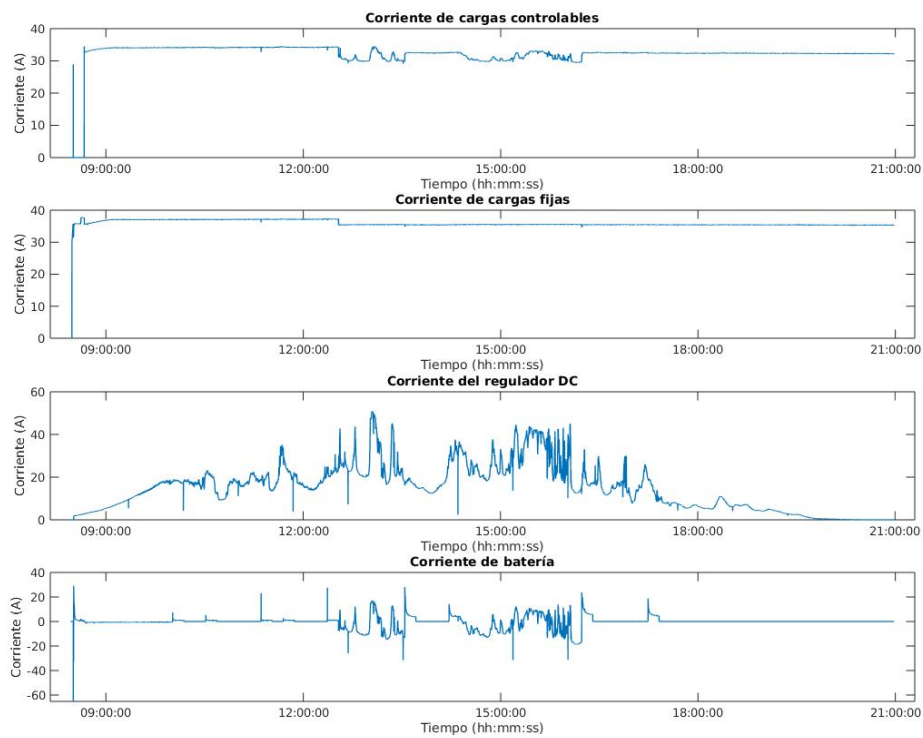


Figura 4.13: Corrientes de la microrred.



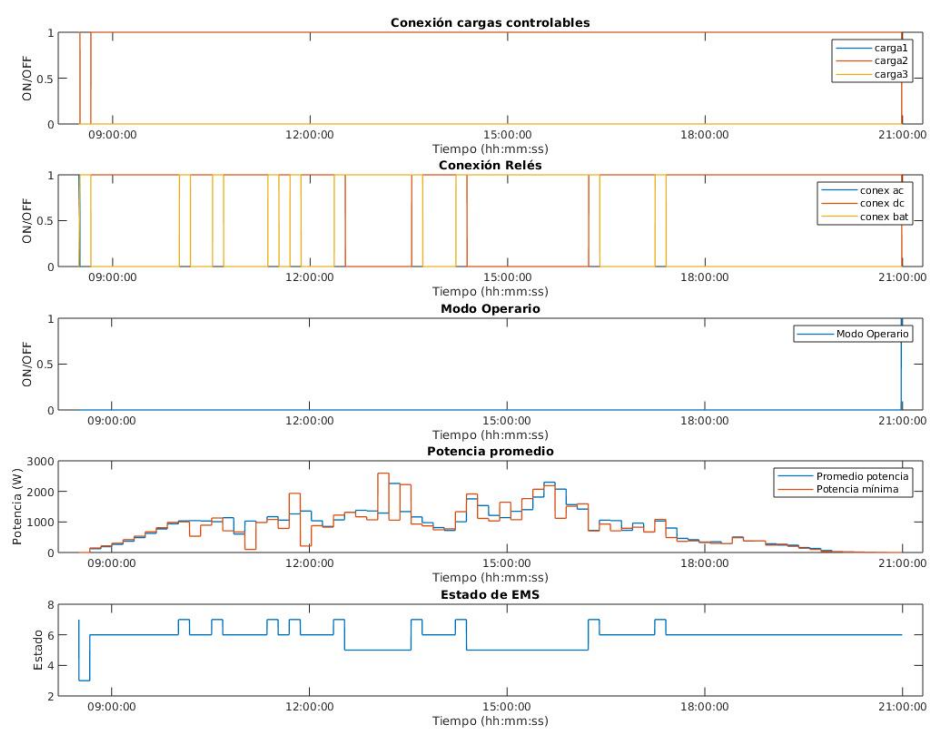


Figura 4.14: Estados y variables de control del EMS.

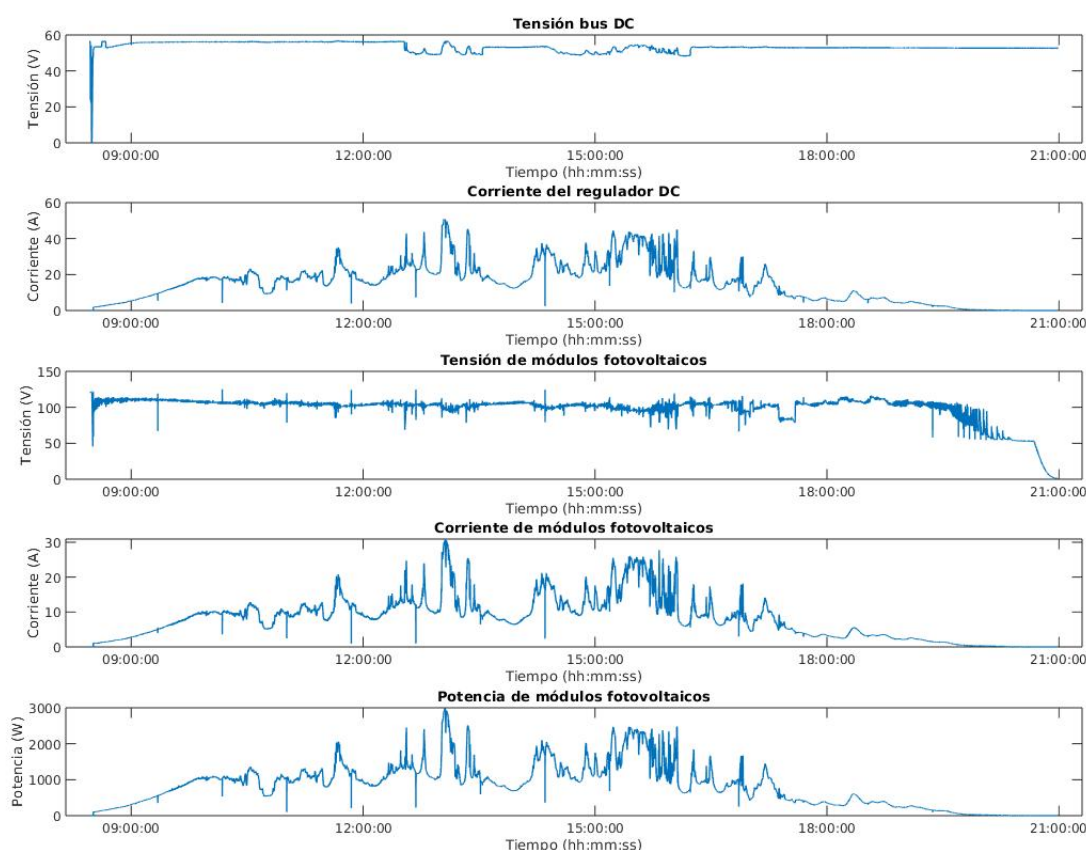


Figura 4.15: Parámetros de los módulos fotovoltaicos.

### 4.3. Pruebas modo operativo

Se han hecho un extenso uso y múltiples pruebas del modo operativo o modo manual en la realización de este trabajo, pero todas ellas no han sido documentadas por encontrarse el trabajo en fases intermedias. En todas ellas se validó el correcto funcionamiento, ya que de lo contrario no se podría haber llevado a cabo la correcta implantación de los diferentes elementos de la microrred.

Las gráficas de la figura 4.16 muestran en detalle la información de las variables de las distintas conexiones y el estado del modo operativo en el momento de finalización de una prueba, en concreto la del día soleado anteriormente mostrado.

Al finalizar una prueba, se activa el modo operativo para tomar el control de los relés y de las cargas. Así, se desconectan primero las cargas controlables (en este caso sólo dos, ya que la tercera no se estaba usando), en segundo lugar el relé de la batería y por último el relé que conecta el bus DC con el equipo de fuerza.

En todo momento, el control de la microrred se hace a través del HMI creado en Node-RED. En este se pueden visualizar todos los parámetros de la microrred al mismo tiempo que se puede operar en ella. En la figura 4.17a se puede ver la pantalla principal del HMI con el modo operativo activo (al igual que la realización de lecturas desde Node-RED) y aparecen en azul los interruptores cerrados y en gris los abiertos (en la imagen no se muestra la secuencia de finalización de pruebas). En la figura 4.17b se

muestra un detalle de la información referida a los módulos fotovoltaicos, en la página secundaria del HMI.

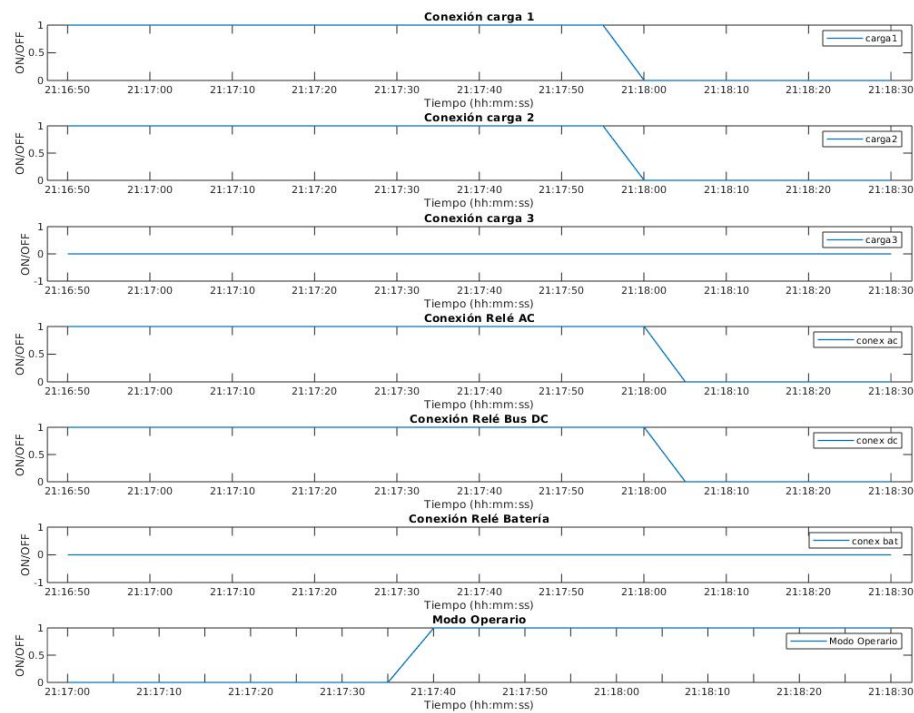
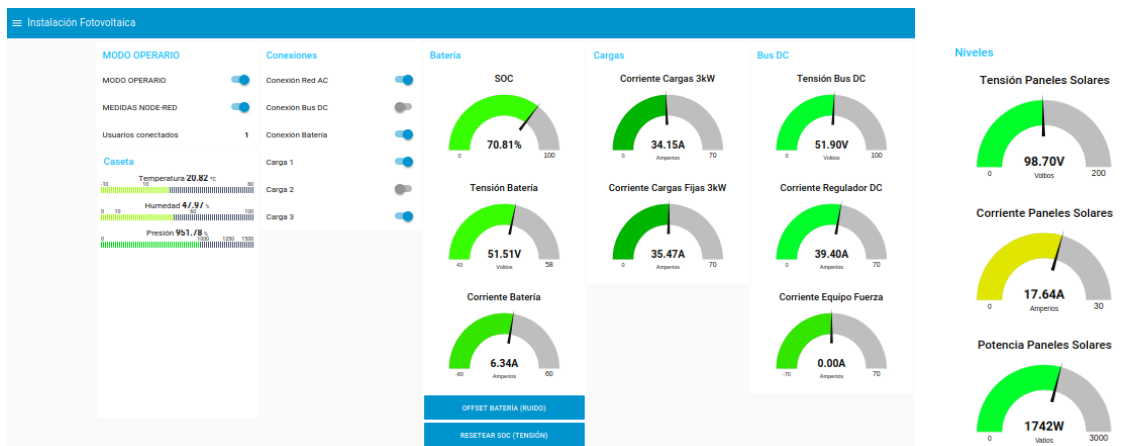


Figura 4.16: Modo operario: Secuencia de finalización de pruebas.



(a) Dashboard de Node-RED: página principal.

(b) Dashboard Node-RED: detalle de información de módulos fotovoltaicos en página secundaria.

Figura 4.17: HMI en modo operario.

## 4.4. Información meteorológica

La información de la estación meteorológica es recogida tanto en la SBC (Raspberry Pi) como en la nube IoT (en este caso se usa la nube de Matlab: ThingSpeak). En este trabajo esta información es meramente informativa y en otros trabajos relacionados sí es usada con propósitos de investigación.

### 4.4.1. Recogida en Controlador DER

En la figura 4.18 se exponen las variables ambientales durante la prueba realizada el 14 de septiembre de 2020 (día nublado). En ella se observan las variaciones suaves de la temperatura ambiente, humedad y presión atmosférica, y la variación más rápida de la temperatura del RTD colocado en uno de los módulos fotovoltaicos. Esta última temperatura aporta mucha información a otros trabajos que buscan estimar la generación de energía fotovoltaica.

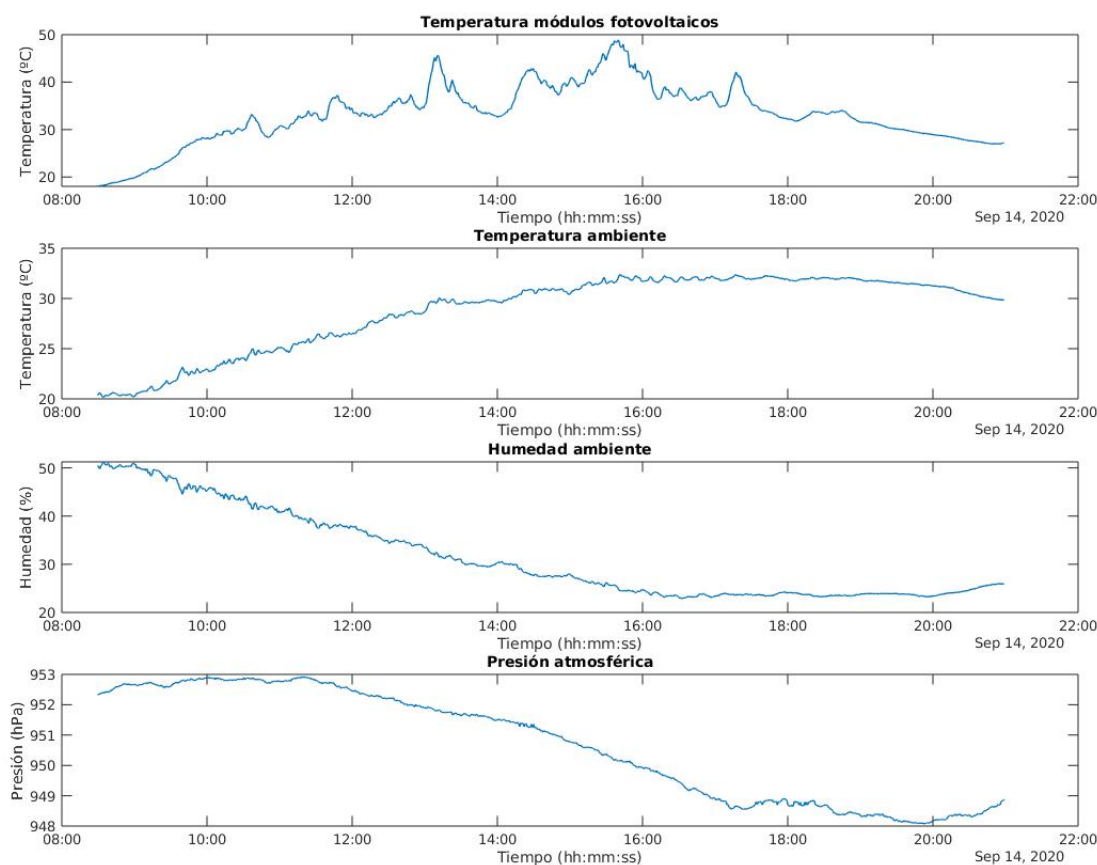


Figura 4.18: Información meteorológica - SBC (Raspberry Pi 4).

### 4.4.2. Recogida en ThingSpeak

Finalmente, en las figuras siguientes (4.19, 4.20 y 4.21), se muestran los datos meteorológicos de varios días (alrededor del 13 de septiembre de 2020) regidos en la nube de Matlab y representados en el mismo *dashboard* que su página *web* proporciona. La ventaja de trabajar con la nube de Matlab es que los datos son fácilmente exportables a Matlab, donde se suele trabajar en el ámbito de la investigación.

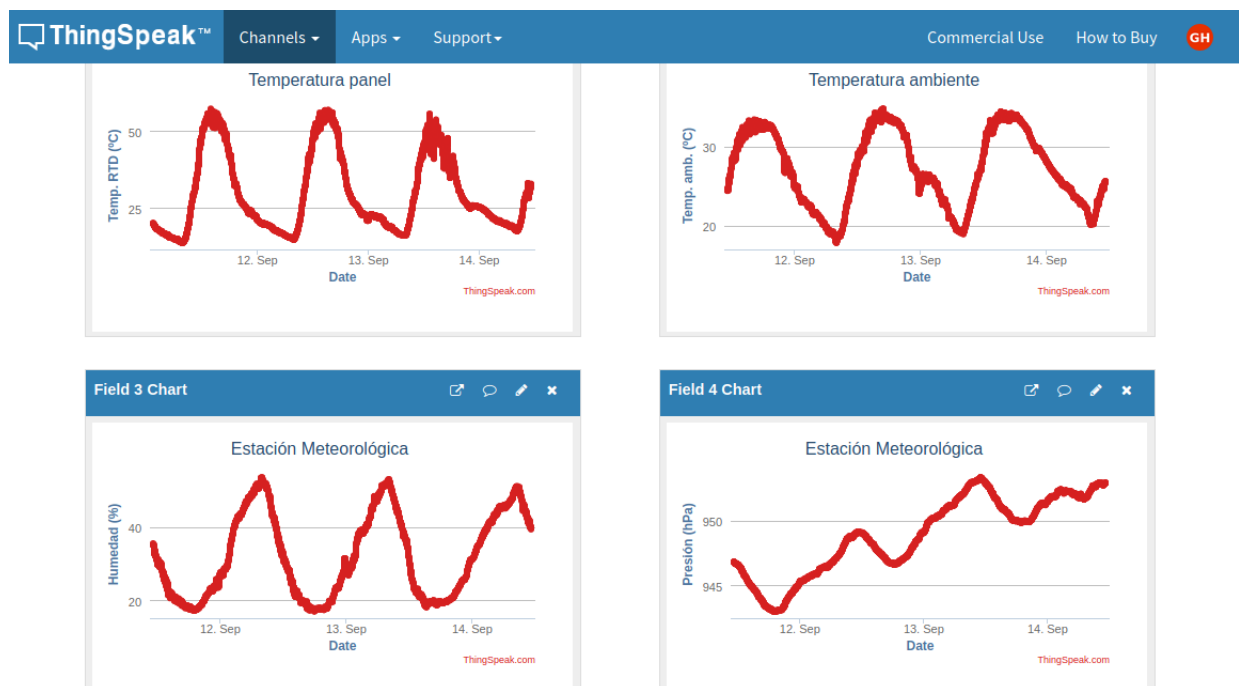


Figura 4.19: Información meteorológica - Nube IoT (ThingSpeak).

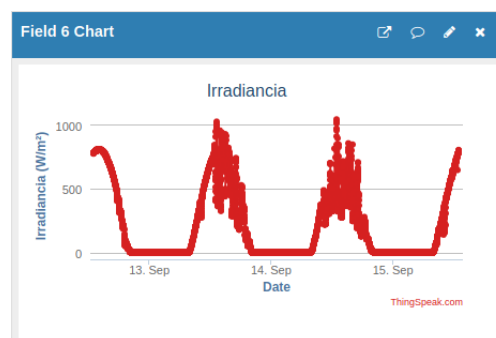


Figura 4.20: Irradiancia - Nube IoT (ThingSpeak).

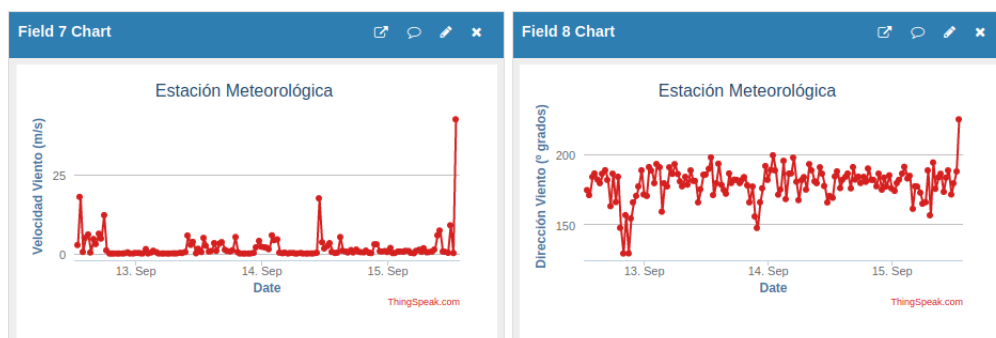


Figura 4.21: Velocidad y dirección del viento - Nube IoT (ThingSpeak).



## Capítulo 5

# Conclusiones y líneas futuras

### 5.1. Conclusiones

El objetivo principal de este trabajo ha sido controlar y gestionar los flujos de energía de una microrred, el cual (como se veía en el capítulo de resultados) ha sido satisfactoriamente alcanzado. Se ha creado un sistema completamente funcional capaz de gestionar la energía producida por una serie de módulos fotovoltaicos con el fin de alimentar diversas cargas eléctricas con el apoyo de un BESS y la red AC.

Como se revisaba al principio de esta memoria, el estado del arte de microrredes en DC se encuentra en un momento de progreso y aumento. Actualmente se están investigando, desarrollando e implementando pequeños prototipos de microrredes en laboratorio, donde se prueban diferentes algoritmos de control de energía (EMS). A diferencia de esos proyectos, en este trabajo se ha creado una microrred real de una potencia superior, usando una instalación fotovoltaica propia y sin usar datos ajenos para simular comportamientos.

En lo que concierne a la implantación del estándar IEC61850, este trabajo se sitúa en el estado del arte, realizando toda la estructura necesaria para su correcto funcionamiento en la microrred. Otros proyectos desarrollan modelos IEC61850 más extensos, pues sus microrredes (prototipos o simulaciones) así lo requieren, pero en lo relacionado con la implementación del estándar y su aplicación práctica se encuentra al mismo nivel.

Por otra parte, con referencia al desarrollo de este trabajo, se debe destacar la importancia de la realización de prototipos para realizar pruebas intermedias y comprobar así el funcionamiento de todos los elementos individualmente. En esta memoria se describen concretamente un par de prototipos de la placa principal del driver hardware y como han sido usados para verificar su correcto funcionamiento en laboratorio, y detectar y solventar problemas con interferencias electromagnéticas producidas por la instalación de la microrred. De esta forma se ha ahorrado mucho tiempo y recursos, ya que la realización de un producto final defectuoso conlleva volver a rediseñarlo entero. Respecto a los problemas con las interferencias electromagnéticas cabe destacar que son un fenómeno muy presente en instalaciones aún siendo de baja potencia. Por esta razón es un aspecto importante a incluir dentro del diseño de electrónica destinada a la gestión de microrredes.

Otro aspecto significativo dentro del desarrollo de este trabajo es el uso de una buena base sobre la cual construir un proyecto. Esto se puede aplicar a cualquier ámbito, pero en este caso está referido a la estructura software del trabajo. En fases iniciales del desarrollo se optó por crear un pequeño sistema de comunicaciones entre diferentes procesos dentro de la SBC (Raspberry Pi) cuando no era un propósito del trabajo. Resultó desfavorablemente y comprometió el resto del trabajo. Finalmente se implementó

ROS como base para las comunicaciones de los distintos procesos, pudiéndose concluir que un buen uso de trabajos previos es tanto igual de útil como legítimo.

En conclusión, todos los objetivos que se plantearon al comienzo de este trabajo han sido satisfactoriamente cumplimentados. Se ha conseguido diseñar e implementar un controlador para una microrred y toda la electrónica implicada en ello. Además se ha implementado el estándar IEC61850, tanto en el procesamiento de la información de la planta como en la gestión del EMS y en sus comunicaciones con clientes exteriores. Asimismo se ha desarrollado un HMI con el que interactuar manualmente con la microrred de forma segura y una estación meteorológica que es de ayuda en otros trabajos actualmente en realización.

## 5.2. Líneas futuras

Este trabajo da pie a diversas líneas de investigación y de realización de proyectos futuros. En primer lugar, un trabajo futuro inmediato podría ser la ampliación de la microrred para que tuviera la posibilidad de inyección de energía a la red AC. Se debería sustituir el equipo de fuerza que actúa como rectificador por un equipo inversor/rectificador. El control de este inversor y los nuevos flujos de energía que aparecerían serían materia extensa para el desarrollo de un nuevo EMS.

En segundo lugar, sería interesante la replicación o creación de alguna microrred como la desarrollada en este trabajo y conectarlas entre sí, para investigar distintas técnicas de control descentralizado en un ámbito real.

En tercer lugar, y relacionado con lo anterior, otro trabajo dentro del estado del arte actual sería el diseño de algoritmos más complejos para EMSs dentro de microrredes más extensas, con BMSs incorporados para una gestión óptima de baterías. Todo ello haciendo uso de la información meteorológica para estimar, por ejemplo, la potencia fotovoltaica a generar en un periodo de tiempo determinado.

Por último, un trabajo más enfocado a las tecnologías IoT podría ser la integración de las microrredes en las *smarthomes* (a través de plataformas como ROS). Así se podría controlar múltiples sistemas de una vivienda de diferentes modos en función de la energía producida o de la demanda de microrredes próximas, entre otros.



## Capítulo 6

# Presupuesto

En este apartado se detalla el presupuesto de ejecución del TFM, componiéndose de dos partes:

- Coste para el desarrollo del diseño hardware y software de la microrred.
- Coste de personal necesario para el desarrollo del trabajo.

La duración de este Trabajo Fin de Máster ha sido de seis meses, que representa el trabajo de un graduado en ingeniería industrial y automática industrial a tiempo completo.



## 6.1. Coste del diseño hardware y software

	Concepto	Cantidad	Precio und. (€)	Total (€)
<b>Hardware</b>	Ordenador Antec	1	1.795,40 €	1.795,40 €
	Monitor Samsung	1	100,00 €	100,00 €
	Teclado y ratón inalámbricos	1	32,90 €	32,90 €
	Raspberry Pi 4 Model B	1	59,26 €	59,26 €
	Adafruit Feather ESP32	1	19,95 €	19,95 €
	ADS1115 16-bit ADC – 4 canales	2	14,95 €	29,90 €
	Expansor I2C PCF8574AN	1	0,50 €	0,50 €
	Adafruit BME680 Environmental sensor	2	18,95 €	37,90 €
	Sensores de corriente (Unidireccional) ACS758LCB-100U-PFF-T	2	6,02 €	12,04 €
	Sensores de corriente (Bidireccional) ACS758LCB-100B-PFF-T	1	6,02 €	6,02 €
	Relé JTN1S-TMP-F-DC12V	1	3,80 €	3,80 €
	Relé EW60-1A3 12VDC 60A	2	12,66 €	25,32 €
	Mosfets - CSD18536KCS 60 V N-Channel	8	3,71 €	29,68 €
	Drivers TC4426	4	1,01 €	4,04 €
	LM-7805	2	0,40 €	0,80 €
	Condensador electrolítico 200uF	4	0,20 €	0,80 €
	Condensador electrolítico 4.7uF	5	0,10 €	0,50 €
	Condensador cerámico 0.33uF	1	0,10 €	0,10 €
	Condensador cerámico 0.1uF	5	0,10 €	0,50 €
	Diodos 1N4001	5	0,14 €	0,70 €
	Resistencias 22K y 330K	1	0,20 €	0,20 €
	RTD pt100	1	11,95 €	11,95 €
	MAX31865	1	14,95 €	14,95 €
	Anemómetro Davis Instruments	1	155,00 €	155,00 €
	Piranómetro SR05	1	200,00 €	200,00 €
	Zócalos	1	2,00 €	2,00 €
	Disipadores	3	5,00 €	15,00 €
	Clemas potencia	1	2,00 €	2,00 €
	Clemas	1	2,00 €	2,00 €
	Placa de prototipado	1	3,00 €	3,00 €
	Cable 60A (metros)	6	1,45 €	8,70 €
<b>Software</b>	Sistema Operativo “Ubuntu”	1	0,00 €	0,00 €
	Sistema Operativo “Raspberry Pi OS”	1	0,00 €	0,00 €
	Node-RED	1	0,00 €	0,00 €
	MATLAB 2019	1	0,00 €	0,00 €
	Nube MATLAB (ThingSpeak)	1	0,00 €	0,00 €
	IDE Arduino	1	0,00 €	0,00 €
	FreeRTOS	1	0,00 €	0,00 €
	Distribución de ROS Kinetic / Melodic	1	0,00 €	0,00 €
	KiCad (Diseño electrónico)	1	0,00 €	0,00 €
<b>TOTAL</b>				<b>2.574,91 €</b>

Tabla 6.1: Coste de los materiales para el diseño hardware y software.

## 6.2. Coste del personal

A continuación se desglosan los sueldos base por hora trabajada de las personas necesarias para realizar este trabajo, incluyendo el coste total que producen estas personas. Se considera que se ha trabajado 37.5 horas semanales y se ha estimado que el mes tiene una media de 22 días laborables. De esta forma, para un total de seis meses, el coste de personal ha sido:

Concepto	Horas	Precio por hora (€)	Total (€)
Graduado en ingeniería industrial	990	11,40 €	11.286,00 €
<b>TOTAL</b>			<b>11.286,00 €</b>

Tabla 6.2: Coste del personal.

## 6.3. Honorarios Facultativos

Estos gastos se han establecido en un 0,255 % sobre el total (costes de material y personal en este caso), según el Colegio Oficial de Ingenieros Técnicos Industriales de Madrid, ya que el coste del proyecto no sobrepasa los 30.050€.

Coste de material y personal	<b>13.860,91 €</b>
Gastos por honorarios facultativos	<b>35,34 €</b>

Tabla 6.3: Coste de honorarios facultativos.

## 6.4. Presupuesto total

Concepto	Precio por hora (€)
Coste del diseño hardware y software	2.574,91 €
Coste del personal	11.286,00 €
Coste de honorarios facultativos	35,34 €
<b>TOTAL (sin IVA)</b>	<b>13.896,25 €</b>
<b>TOTAL (con IVA 21 %)</b>	<b>16.814,46 €</b>

Tabla 6.4: Coste total del trabajo.

El presupuesto total del proyecto asciende a trece mil ochocientos noventa y seis euros con veinticinco céntimos. Teniendo en cuenta la carga impositiva del 21 % por el IVA, a fecha de octubre del año 2020, el coste total del trabajo ascendería a dieciséis mil ochocientos catorce euros con cuarenta y seis céntimos.

# Bibliografía

- [1] E. M. Gui, M. Diesendorf, and I. MacGill, “Distributed energy infrastructure paradigm: Community microgrids in a new institutional economics context,” *Renewable and Sustainable Energy Reviews*, vol. 72, pp. 1355–1365, 2017.
- [2] B. Mataloto, J. C. Ferreira, and N. Cruz, “Lobems-iot for building and energy management systems,” *Electronics*, vol. 8, no. 7, p. 763, 2019.
- [3] A. M. Fadel, A. M. Ibrahim, A. K. Tamazin, R. A. Hamdy, and A. S. Abdel-Khalik, “Iot-based power management for dc microgrids,” in *2019 21st International Middle East Power Systems Conference (MEPCON)*. IEEE, 2019, pp. 1107–1111.
- [4] A. Elmouatamid, Y. NaitMalek, M. Bakhouya, R. Ouladsine, N. Elkamoun, K. Zine-Dine, and M. Khaidar, “An energy management platform for micro-grid systems using internet of things and big-data technologies,” *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 233, no. 7, pp. 904–917, 2019.
- [5] M. Boussetta, S. Motahhir, R. El Bachtiri, A. Allouhi, M. Khanfara, and Y. Chaibi, “Design and embedded implementation of a power management controller for wind-pv-diesel microgrid system,” *International Journal of Photoenergy*, vol. 2019, 2019.
- [6] A. Apostolov, “Integration of distributed energy resources in electric power systems,” *Energize RE Renew. Energy Suppl*, pp. 1–12, 2015.
- [7] A. Salazar-Zuluaga, L. A. Aguilar-Londoño, G. D. Zapata-Madrigal, and R. García-Sierra, “Implementación de nodos lógicos der iec 61850-7-420 en una placa electrónica,” *TecnoLógicas*, vol. 20, no. 40, pp. 13–28, 2017.
- [8] T. S. Ustun and S. Hussain, “Secure communication modeling for microgrid energy management system: Development and application,” *Energies*, vol. 13, no. 1, p. 68, 2020.
- [9] G. Gutiérrez-Alcaraz, E. Galván, N. González-Cabrera, and M. Javadi, “Renewable energy resources short-term scheduling and dynamic network reconfiguration for sustainable energy consumption,” *Renewable and Sustainable Energy Reviews*, 2015. [Online]. Available: <https://ideas.repec.org/a/eee/rensus/v52y2015icp256-264.html>
- [10] Z. Li, S. Bahramirad, A. Paaso, M. Yan, and M. Shahidehpour, “Blockchain for decentralized transactive energy management system in networked microgrids,” *The Electricity Journal*, vol. 32, no. 4, pp. 58 – 72, 2019, special Issue on Strategies for a sustainable, reliable and resilient grid. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S104061901930048X>
- [11] F. Nadeem, S. S. Hussain, P. K. Tiwari, A. K. Goswami, and T. S. Ustun, “Comparative review of energy storage systems, their roles, and impacts on future power systems,” *IEEE Access*, vol. 7, pp. 4555–4585, 2018.

- [12] “Tp672p - 320/325/330w polycrystalline solar - module72 cell series,” [https://www.talesunenergy.com/fileadmin/user\\_upload/pdf/data\\_sheets/TP672P.pdf](https://www.talesunenergy.com/fileadmin/user_upload/pdf/data_sheets/TP672P.pdf), accessed: 2020-08-31.
- [13] “Bauer - regulador de carga mpptsr-ml 12/24/36/48v 60a,” <https://autosolar.es/pdf/manual-regulador-mppt-bauer-12-24-36-48-60a.pdf>, accessed: 2020-09-01.
- [14] “Battery performance specifications - 12v125f,” <https://datasheet.octopart.com/12V125F-EnerSys-datasheet-62067597.pdf>, accessed: 2020-09-01.
- [15] “Huawei - apm30h,” <https://fccid.io/ANATEL/03515-13-03257/Manual/51E9F29C-1DCE-4A94-8275-1A7E1FF7533A>, accessed: 2020-09-01.
- [16] G. La Tona, M. Luna, A. Di Piazza, and M. C. Di Piazza, “Towards the real-world deployment of a smart home ems: A dp implementation on the raspberry pi,” *Applied Sciences*, vol. 9, no. 10, p. 2120, 2019.
- [17] “Indacar - internet de las cosas y movilidad,” <http://blog.indacar.io/index.php/2018/04/22/internet-de-las-cosas-y-movilidad/>, accessed: 2020-10-08.
- [18] “About us - raspberry pi foundation,” <https://www.raspberrypi.org/about/>, accessed: 2020-08-26.
- [19] “Bigtronica - raspberry pi 4 model b,” <https://www.bigtronica.com/poblado/sistemas-mini-pc/raspberry-pi/2204-raspberry-pi-4-modelo-b-4-gb-ram.html>, accessed: 2020-10-08.
- [20] D. Rwegasira, N. Mvungi, and H. Tenhunen, “Agent based system for improved control and monitoring of a solar driven dc microgrid,” in *2018 IEEE PES/IAS PowerAfrica*. IEEE, 2018, pp. 890–894.
- [21] A. Iovine, T. Rigaut, G. Damm, E. De Santis, and M. D. Di Benedetto, “Power management for a dc microgrid integrating renewables and storages,” *Control Engineering Practice*, vol. 85, pp. 59–79, 2019.
- [22] D. Rwegarsira, I. B. Dhaou, A. Kondoro, A. Kelati, N. Mvungi, and H. Tenhunen, “A hardware-in-loop simulation of dc microgrid using multi-agent systems,” in *2018 22nd Conference of Open Innovations Association (FRUCT)*. IEEE, 2018, pp. 232–237.
- [23] “Esp32 - espressif,” [https://www.espressif.com/sites/default/files/documentation/esp32\\_technical\\_reference\\_manual\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf), accessed: 2020-08-27.
- [24] L. Van Der Zel, “Guidelines for implementing substation automation using iec61850, the international power system information modeling standard,” *Technical Report*, 2004.
- [25] Y. Rangelov, N. Nikolaev, and M. Ivanova, “The iec 61850 standard - communication networks and automation systems from an electrical engineering point of view,” in *2016 19th International Symposium on Electrical Apparatus and Technologies (SIELA)*, 2016, pp. 1–4.
- [26] S. M. S. Hussain, A. Tak, T. S. Ustun, and I. Ali, “Communication modeling of solar home system and smart meter in smart grids,” *IEEE Access*, vol. 6, pp. 16 985–16 996, 2018.
- [27] S. Marzal, R. Salas, R. González-Medina, G. Garcerá, and E. Figueres, “Current challenges and future trends in the field of communication architectures for microgrids,” *Renewable and Sustainable Energy Reviews*, vol. 82, pp. 3610–3622, 2018.
- [28] R. E. Mackiewicz, “Overview of iec 61850 and benefits,” in *2006 IEEE Power Engineering Society General Meeting*. IEEE, 2006, pp. 8–pp.

- [29] “Allegro microsystems,” <https://www.allegromicro.com>, accessed: 2020-08-30.
- [30] “Texas instruments - pcf8574aremot8-bit i/o expanderfor i2c bus,” [https://www.ti.com/lit/ds/symlink/pcf8574a.pdf?ts=1598804536548&ref\\_url=https%253A%252F%252Fwww.google.com%252F](https://www.ti.com/lit/ds/symlink/pcf8574a.pdf?ts=1598804536548&ref_url=https%253A%252F%252Fwww.google.com%252F), accessed: 2020-08-30.
- [31] “Datasheet - stmicroelectronics,” <https://www.st.com>, accessed: 2020-08-30.
- [32] “Microchip - 1.5a dual high-speed power mosfet drivers,” <http://ww1.microchip.com/downloads/en/devicedoc/20001422g.pdf>, accessed: 2020-08-30.
- [33] “Texas instruments - sd18536kcs60 v n-channelnxfet powermosfet,” [https://www.ti.com/lit/ds/slps532a/slps532a.pdf?ts=1598804855050&ref\\_url=https%253A%252F%252Fwww.google.com%252F](https://www.ti.com/lit/ds/slps532a/slps532a.pdf?ts=1598804855050&ref_url=https%253A%252F%252Fwww.google.com%252F), accessed: 2020-08-30.
- [34] “Mouser - jtn1as-tmp-f-dc12v,” <https://www.mouser.es/ProductDetail/Panasonic-Industrial-Devices/JTN1AS-TMP-F-DC12V?qs=UO%2Fx91QLkSB1OQd9YHkmww=>, accessed: 2020-08-30.
- [35] “Ads1115 - texas instruments,” <http://www.ti.com/lit/ds/symlink/ads1114.pdf>, accessed: 2020-08-15.
- [36] “Bme680 - bosch,” <https://cdn-shop.adafruit.com/product-files/3660/BME680.pdf>, accessed: 2020-08-15.
- [37] “Te connectivity - power latching relay ew60,” [https://www.te.com/commerce/DocumentDelivery/DDEController?Action=showdoc&DocId=Data+Sheet%7FEW60\\_Datasheet\\_07042016%7F0416%7Fpdf%7FEnglish%7FENG\\_DS\\_EW60\\_Datasheet\\_07042016\\_0416.pdf%7F2071366-1](https://www.te.com/commerce/DocumentDelivery/DDEController?Action=showdoc&DocId=Data+Sheet%7FEW60_Datasheet_07042016%7F0416%7Fpdf%7FEnglish%7FENG_DS_EW60_Datasheet_07042016_0416.pdf%7F2071366-1), accessed: 2020-08-30.
- [38] F. J. Meca, *Acondicionamiento de señales y adquisición de datos (MUIE)*. UAH, 2019, ch. Bloque 2 - Interferencias electromagnéticas.
- [39] “Wikipedia - file:emc noise model.png,” [https://en.wikipedia.org/wiki/File:EMC\\_Noise\\_Model.png](https://en.wikipedia.org/wiki/File:EMC_Noise_Model.png), accessed: 2020-10-08.
- [40] J. Golten and A. Verwer, *Control System Design and Simulation*. McGraw-Hill Publishing Company, 1991, ch. Chapter 3 - System Time Response, pp. 71 – 72.
- [41] “Datasheet raspberry pi 4 model b,” [https://www.raspberrypi.org/documentation/hardware/raspberrypi/bcm2711/rpi\\_DATA\\_2711\\_1p0\\_preliminary.pdf](https://www.raspberrypi.org/documentation/hardware/raspberrypi/bcm2711/rpi_DATA_2711_1p0_preliminary.pdf), accessed: 2020-08-30.
- [42] “Mqtt: The standard for iot messaging,” <https://mqtt.org/>, accessed: 2020-08-30.
- [43] “Mqtt org,” <https://mqtt.org/mqtt-specification/>, accessed: 2020-10-08.
- [44] “Hukseflux - user manual sr05 series,” [https://www.hukseflux.com/uploads/product-documents/SR05\\_digital\\_series\\_manual\\_v1814.pdf](https://www.hukseflux.com/uploads/product-documents/SR05_digital_series_manual_v1814.pdf), accessed: 2020-09-01.
- [45] “Anemometer - davis instruments,” [https://www.davisinstruments.com/product\\_documents/weather/spec\\_sheets/7911\\_SS.pdf](https://www.davisinstruments.com/product_documents/weather/spec_sheets/7911_SS.pdf), accessed: 2020-09-01.
- [46] “Maxim integrated - max31865,” <https://datasheets.maximintegrated.com/en/ds/MAX31865.pdf>, accessed: 2020-09-01.

- [47] “Adafruit huzzah32 - esp32 feather,” <https://cdn-learn.adafruit.com/downloads/pdf/adafruit-huzzah32-esp32-feather.pdf>, accessed: 2020-08-30.
- [48] “Freertos org,” <https://www.freertos.org/index.html>, accessed: 2020-10-08.
- [49] “Using distributed energy resources,” <https://web.archive.org/web/20200731133916/https://www.nrel.gov/docs/fy02osti/31570.pdf>, accessed: 2020-08-28.
- [50] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, “Ros: an open-source robot operating system,” in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.
- [51] “Ros org,” <https://www.ros.org/>, accessed: 2020-10-08.
- [52] J. Zabaleta, I. SAN MARTÍN, and J. PASCUAL, “Cálculo del estado de carga en baterías de plomo-ácido: diseño y validación experimental,” *Universidad Pública de Navarra*, 2016.
- [53] B. Drury, *Control Techniques Drives and Controls Handbook*. Institution of Engineering and Technology, 2009, p. 508.
- [54] “Open source libraries for iec 61850 and iec 60870-5-104,” <https://libiec61850.com/libiec61850/new-release-1-4-2-of-libiec61850/>, accessed: 2020-08-29.
- [55] “Node-red,” <https://nodered.org/>, accessed: 2020-08-15.
- [56] “Aemet - resumen mensual de la radiación solar. septiembre 2015.” [https://www.aemet.es/documentos/es/serviciosclimaticos/vigilancia\\_clima/radiacion\\_ozono/radiacion\\_solar/2015/InformeRad\\_Solar\\_2015\\_09.pdf](https://www.aemet.es/documentos/es/serviciosclimaticos/vigilancia_clima/radiacion_ozono/radiacion_solar/2015/InformeRad_Solar_2015_09.pdf), p. 9, 2015, accessed: 2020-10-05.



## **Apéndice A**

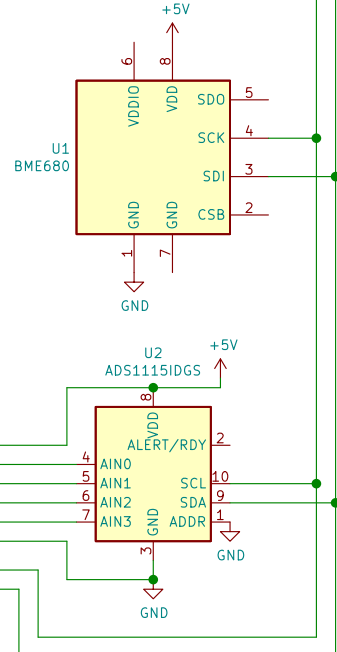
# **Esquemáticos del Driver Hardware**

Sheet: placa\_principal

File: placa\_principal.sch

Vcc\_ADC  
Vout\_corriente\_bateria  
Vout\_corriente\_cargas\_controlables  
Vout\_corriente\_cargas\_fijas  
Vout\_tension\_bateria\_acond  
GND\_ADC  
SCL  
SDA  
Rele2\_T1D  
Rele2\_T2D  
Rele2\_T3D  
Rele1\_T1D  
Rele1\_T2D  
Rele1\_T3D

J1  
Screw\_Terminal\_01x01  
J2  
Screw\_Terminal\_01x01  
J3  
Screw\_Terminal\_01x01



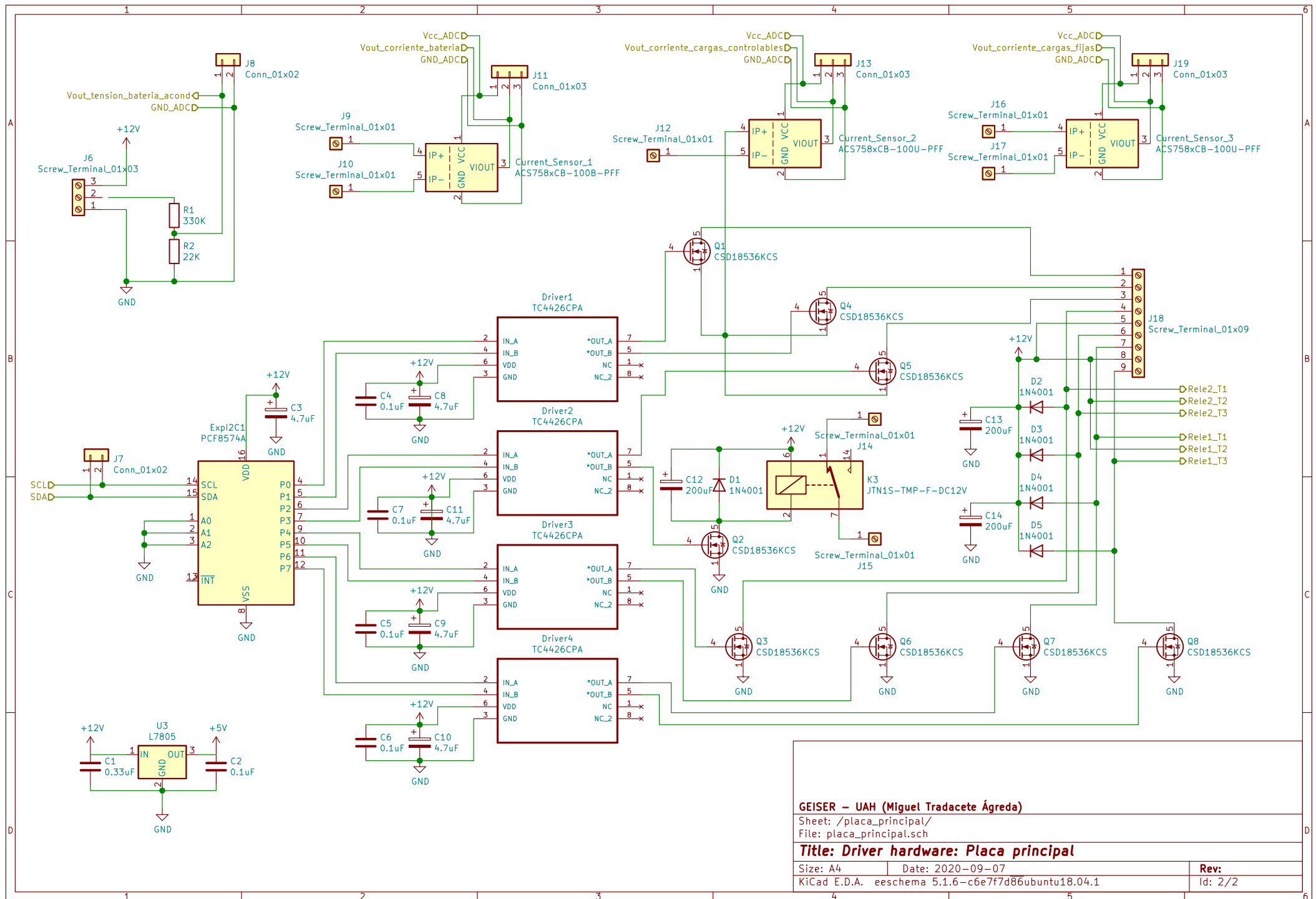
GEISER – UAH (Miguel Tradacete Ágreda)

Sheet: /  
File: driver\_hardware.sch

Title: Driver hardware

Size: A4  
Date: 2020-09-07  
KiCad E.D.A. eeschema 5.1.6-c6e7f7d86ubuntu18.04.1

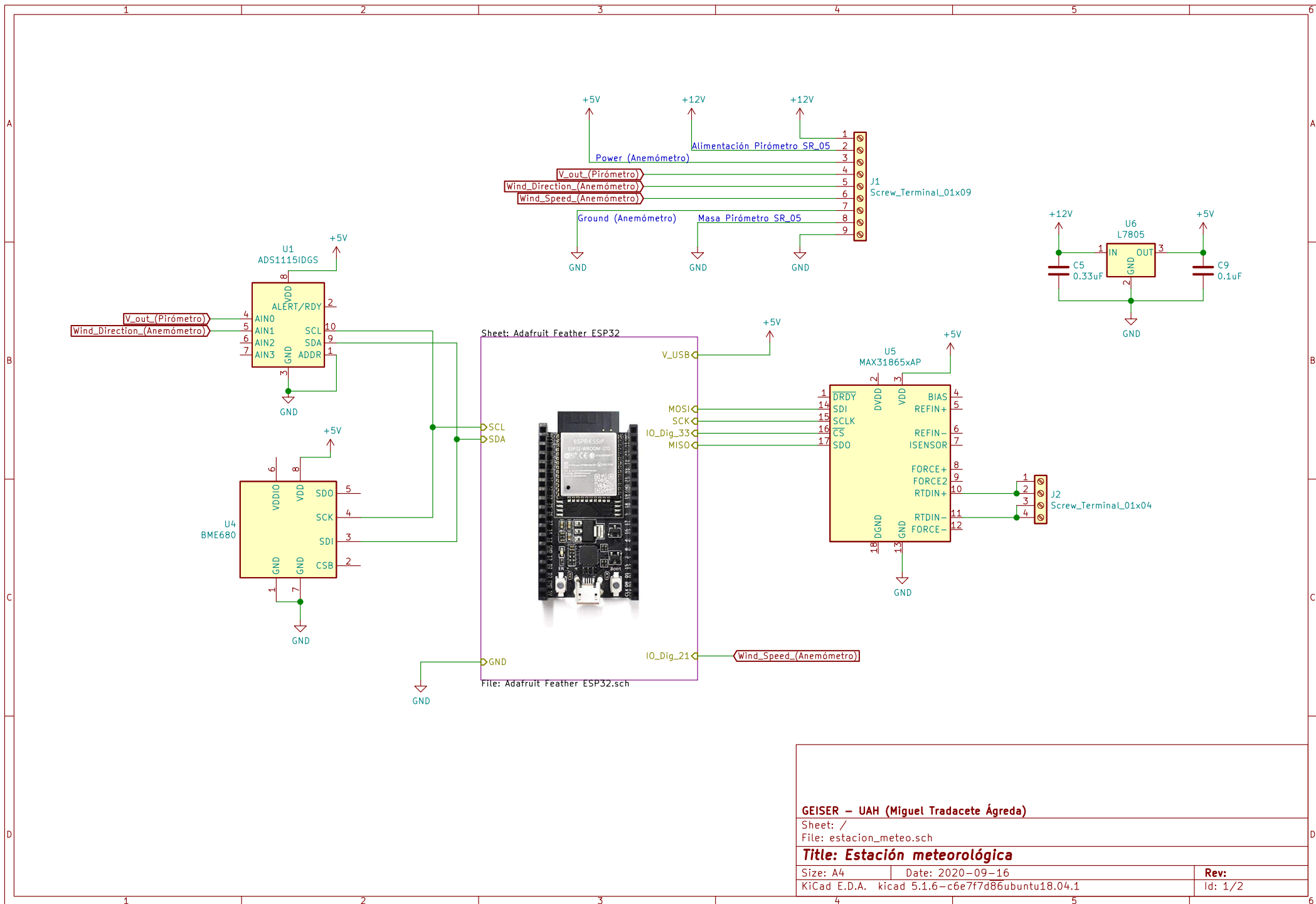
Rev:  
Id: 1/2





## Apéndice B

# Esquemáticos de la Estación Meteorológica



GEISER – UAH (Miguel Tradacete Ágreda)

Sheet: /  
File: estacion\_meteo.sch

**Title: Estación meteorológica**

Size: A4  
KiCad E.D.A. kicad 5.1.6-c6e7f7d86ubuntu18.04.1

Date: 2020-09-16

Rev:  
Id: 1/2







## Apéndice C

### Modelo IEC61850 de la planta de la microrred

# MODELO PLANTA FOTOVOLTAICA HELIOS

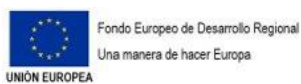
## IEC 61850

# 1. DEFINICIÓN DE LOS DISPOSITIVO LÓGICOS

- **LD MNGT:** En este dispositivo lógico se engloban todas las funcionalidades necesarias para la gestión del modo de operación de la planta global. Algunas de las funcionalidades son:
  - Información y definición de los parámetros de gestión de la planta.
  - Apertura y cierre de los diferentes interruptores de la planta, para que trabaje en los diferentes modos en función de la generación solar y el precio de la electricidad.
- **LD ACGRID:** Este dispositivo lógico engloba todo lo necesario para la conexión de la planta con la red AC. En este caso
  - Rectificador
- **LD PV:** En este dispositivo lógico se engloban todas las características y toda la información sobre la generación mediante las placas fotovoltaicas:
  - Información sobre la matriz de paneles instalada.
  - Información sobre el regulador MPPT
- **LD BESS:** Sistema de almacenamiento de energía en baterías:
  - Información sobre los parámetros más característicos del sistema baterías usado.
- **LD PLOAD:** Dispositivo lógico donde se recoge información de las cargas prioritarias conectadas en el sistema. Principalmente información sobre variables eléctricas.
- **LD NPLOAD:** Dispositivo lógico donde se recoge información de las cargas no prioritarias conectadas en el sistema. Principalmente información sobre variables eléctricas.
- **LD WEATHERSTATION:** Dispositivo lógico donde se recoge la información relevante sobre las variables del tiempo:
  - Temperatura
  - Humedad
  - Viento
  - Irradiancia

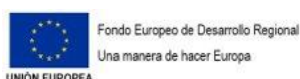


## 2.ASIGNACIÓN DE LOS NODOS LÓGICOS A LOS DISPOSITIVOS LÓGICOS





Logical Device (LD)	Logical Node (LN)	Explanation
LD Management	LLNO	Contiene información de los dispositivos lógicos
	LPHD	Contiene información de los dispositivos físicos
	DCRP	Características corporativas de la planta DER en el ECP. Incluye: Propiedad, operación, autoría, obligaciones contractuales y permisos, ubicación e identificadores de todos los dispositivos DER conectados directa o indirectamente en el ECP.
	DOPR	Características operacionales en el ECP, incluye tipo de dispositivos DER, tipo de conexión, modos de operación, clasificación combinada de todas las DER en el ECP, límites de operación del sistema de potencia en el ECP.
	DOPM	Modo de operación del DER en el punto de conexión. Este LN puede ser usado para fijar los modos de operación disponibles junto con los modos de operación actuales.
	DPST	Estado actual en cada ECP, incluye estado de conexión de la planta DER, alarmas.
	DCCT	Parámetros de despacho económico para las operaciones DER
	DSCC	Control de los programas de energía y servicios auxiliares
	DSCH	Programa de la planta DER para proporcionar energía o los servicios auxiliares
	CSWI1	Control del interruptor → Relé 1
	XCBR1	Interruptor de circuito → Relé 1
	CSWI2	Control del interruptor → Relé 2
	XCBR2	Interruptor de circuito → Relé 2
	CSWI3	Control del interruptor → Relé 3
	XCBR3	Interruptor de circuito → Relé 3
	CSWI4	Control del interruptor → Mosfet 1 cargas no prioritarias
	XCBR4	Interruptor de circuito → Mosfet 1 cargas no prioritarias
	CSWI5	Control del interruptor → Mosfet 2 cargas no prioritarias

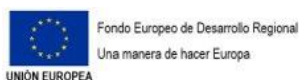




	<b>XCBR5</b>	Interruptor de circuito → Mosfet 2 cargas no prioritarias
	<b>CSWI6</b>	Control del interruptor → Mosfet 3 cargas no prioritarias
	<b>XCBR6</b>	Interruptor de circuito → Mosfet 3 cargas no prioritarias

Logical Device (LD)	Logical Node (LN)	Explanation
LD PV	<b>LLNO</b>	Contiene información de los dispositivos lógicos
	<b>LPHD</b>	Contiene información de los dispositivos físicos
	<b>DPVM</b>	Información sobre el módulo PV
	<b>DPVA</b>	Características del array PV. Ofrece información general del array PV
	<b>DPVC</b>	Control de la matriz PV. Usado para maximizar la potencia de salida de la matriz.
	<b>MMDC</b>	Medidas de las variables eléctricas DC

Logical Device (LD)	Logical Node (LN)	Explanation
LD BESS	<b>LLNO</b>	Contiene información de los dispositivos lógicos
	<b>LPHD</b>	Contiene información de los dispositivos físicos
	<b>ZBAT</b>	Características del sistema de baterías
	<b>ZBTC</b>	Carga del sistema de baterías
	<b>MMDC</b>	Medidas de las variables eléctricas DC

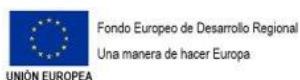




Logical Device (LD)	Logical Node (LN)	Explanation
LD PLOAD	LLN0	Contiene información de los dispositivos lógicos
	LPHD	Contiene información de los dispositivos físicos
	MMDC	Medidas de las variables eléctricas DC

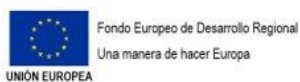
Logical Device (LD)	Logical Node (LN)	Explanation
LD NPLOAD	LLN0	Contiene información de los dispositivos lógicos
	LPHD	Contiene información de los dispositivos físicos
	MMDC	Medidas de las variables eléctricas DC

Logical Device (LD)	Logical Node (LN)	Explanation
LD ACGRID	LLN0	Contiene información de los dispositivos lógicos
	LPHD	Contiene información de los dispositivos físicos
	ZRCT	Características del rectificador





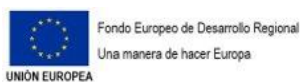
Logical Device (LD)	Logical Node (LN)	Explanation
LD WeatherStation	LLNO	Contiene información de los dispositivos lógicos
	LPHD	Contiene información de los dispositivos físicos
	MMET	Información meteorológica





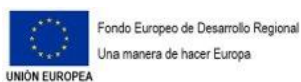


### 3.ASIGNACIÓN DE LOS OBJETOS DE DATOS A LOS NODOS LÓGICOS



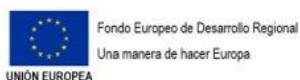


Logical Node (LN)	Data Object (DO)	Explanation	Value
LN DCRP	NamPlt	Placa de identificación	
	Beh	Comportamiento	
	Health	Salud (estado)	
	Mod	Modo	
	PlntObSelf	Propósito/obligaciones de la planta en el ECP – True = funciona de forma pasiva o siempre que sea posible (fotovoltaica)	1
	PlntObBck	Propósito/obligaciones en el ECP – True = Para respaldo (backuo)	0





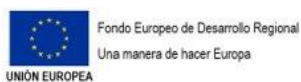
Logical Node (LN)	Data Object (DO)	Explanation	Value	
LN DOPR	NamPlt	Placa de identificación		
	Beh	Comportamiento		
	Health	Salud (estado)		
	Mod	Modo		
	ECPTtype	Tipo de ECP:		3
		Value	Explanation	
		0	No aplicable / Desconocido	
		1	Conexión de una única DER a una carga local	
		2	Conexión del grupo de DERs al EPS local, sirviendo una carga local	
		3	Conexión del EPS local con la carga local al EPS de la zona (PCC)	
		4	Conexión del EPS local sin carga local al EPS de la zona (PCC)	
99	Otro			
ECPID	Identificación del ECP			
ECPNomWRtg	Nominal, mínimo, y máximo agregado de watsios DER en el ECP			
ECPNomVLev	Nominal, mínimo, y máximo nivel de voltaje en el ECP			





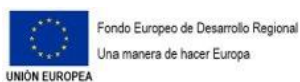
Logical Node (LN)	Data Object (DO)	Explanation	Value
LN DOPM	NamPlt	Placa de identificación	
	Beh	Comportamiento	
	Health	Salud (estado)	
	Mod	Modo	
	OpModPM	Modos de operación – Impulsado por la fuente de energía solar, por lo que el nivel de generación depende de la disponibilidad de esa fuente de energía.	
	OpModConW	Modo de operación – vatios constantes	
	OpModConV	Modo de operación – Voltaje constante	
	OpModExIm	Modo de operación – Exportación/importación constante	
	OpModIsld	Modo de operación – Isla en el ECP	

Logical Node (LN)	Data Object (DO)	Explanation	Value	
LN DOPM	NamPlt	Placa de identificación		
	Beh	Comportamiento		
	Health	Salud (estado)		
	Mod	Modo		
	OpTms	Tiempo de operación desde la puesta en marcha		
	ECPConn	Conexión de la planta DER en el ECP:		1
		Value	Explanation	
		True	Conectado eléctricamente al ECP	
False		No conectado eléctricamente al ECP		
TotWh	Total de watios-hora al ECP desde el último reset			



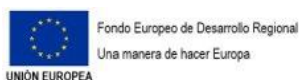


Logical Node (LN)	Data Object (DO)	Explanation	Value
LN DCCT	NamPlt	Placa de identificación	
	Beh	Comportamiento	
	Health	Salud (estado)	
	Mod	Modo	
	Currency	Código de 3 caracteres de la moneda ISO 4217	978
	OpCost	Coste operacional marginal por hora	
	OpWCost	Coste operacional marginal por kWh	
	StrCost	Coste de la puesta en marcha de la DER	
	StopCost	Coste de parada de la DER	





Logical Node (LN)	Data Object (DO)	Explanation	Value
LN DSCC	NamPlt	Placa de identificación	
	Beh	Comportamiento	
	Health	Salud (estado)	
	Mod	Modo	
	ActWSchSt	Indicación de qué programa de energía está activo – Schedule 0 indica no programa	
	ActAncSchdSt	Indicación de que programa de servicios auxiliares está activo – Schedule 0 indica no programa.	
	ActWSchd	Activar un programa de energía específico, usando TimeActivatedOperate para establecer la hora de inicio de los programas usando el tiempo relativo y si la hora de inicio está en el futuro. ctrVal: 0 = desactivar, 1 = activar	
	ActAncSchd	Activar un programa de equipos auxiliares específico, usando TimeActivatedOperate para establecer la hora de inicio de los programas usando el tiempo relativo y si la hora de inicio está en el futuro. ctrVal: 0 = desactivar, 1 = activar	





Logical Node (LN)	Data Object (DO)	Explanation	Value																									
LN DSCH	NamPlt	Placa de identificación																										
	Beh	Comportamiento																										
	Health	Salud (estado)																										
	Mod	Modo																										
	SchdSt	Indica que el programa ha sido activado																										
	SchdId	Identidad no nula del programa																										
	SchdTyp	Tipo de programa, identificando el modo de funcionamiento en el que se utilizará el programa: <table><thead><tr><th>Value</th><th>Explanation</th></tr></thead><tbody><tr><td>0</td><td>No aplicable / Desconocido</td></tr><tr><td>1</td><td>Energía</td></tr><tr><td>2</td><td>Reserva de contingencia “spinning”</td></tr><tr><td>3</td><td>Reserva suplementaria de contingencia</td></tr><tr><td>4</td><td>Reserva de emergencia</td></tr><tr><td>5</td><td>Reserva de emisiones</td></tr><tr><td>6</td><td>Balance energético</td></tr><tr><td>7</td><td>Potencia reactiva</td></tr><tr><td>8</td><td>Black start</td></tr><tr><td>9</td><td>Isla emergencia</td></tr><tr><td>99</td><td>Otro</td></tr></tbody></table>		Value	Explanation	0	No aplicable / Desconocido	1	Energía	2	Reserva de contingencia “spinning”	3	Reserva suplementaria de contingencia	4	Reserva de emergencia	5	Reserva de emisiones	6	Balance energético	7	Potencia reactiva	8	Black start	9	Isla emergencia	99	Otro	1
	Value	Explanation																										
	0	No aplicable / Desconocido																										
1	Energía																											
2	Reserva de contingencia “spinning”																											
3	Reserva suplementaria de contingencia																											
4	Reserva de emergencia																											
5	Reserva de emisiones																											
6	Balance energético																											
7	Potencia reactiva																											
8	Black start																											
9	Isla emergencia																											
99	Otro																											

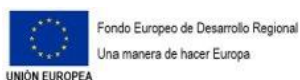


	<b>SchdCat</b>	<table><tr><td colspan="2">Categoría del programa:</td></tr><tr><td>Value</td><td>Explanation</td></tr><tr><td>0</td><td>No aplicable / Desconocido</td></tr><tr><td>1</td><td>Regular</td></tr><tr><td>2</td><td>BackUp</td></tr><tr><td>3</td><td>Emergencia</td></tr><tr><td>4</td><td>Mantenimiento</td></tr><tr><td>99</td><td>Otro</td></tr></table>	Categoría del programa:		Value	Explanation	0	No aplicable / Desconocido	1	Regular	2	BackUp	3	Emergencia	4	Mantenimiento	99	Otro							
Categoría del programa:																									
Value	Explanation																								
0	No aplicable / Desconocido																								
1	Regular																								
2	BackUp																								
3	Emergencia																								
4	Mantenimiento																								
99	Otro																								
	<b>SchdVal</b>	<table><tr><td colspan="2">Significado del parámetro val en el SCA o SCR:</td></tr><tr><td>Value</td><td>Explanation</td></tr><tr><td>0</td><td>No aplicable / Desconocido</td></tr><tr><td>1</td><td>Potencia activa</td></tr><tr><td>2</td><td>Potencia Reactiva</td></tr><tr><td>3</td><td>Factor de potencia</td></tr><tr><td>4</td><td>Voltaje</td></tr><tr><td>5</td><td>Precio por la potencia activa</td></tr><tr><td>6</td><td>Precio por la potencia reactiva</td></tr><tr><td>7</td><td>Calor</td></tr><tr><td>99</td><td>Otro</td></tr></table>	Significado del parámetro val en el SCA o SCR:		Value	Explanation	0	No aplicable / Desconocido	1	Potencia activa	2	Potencia Reactiva	3	Factor de potencia	4	Voltaje	5	Precio por la potencia activa	6	Precio por la potencia reactiva	7	Calor	99	Otro	
Significado del parámetro val en el SCA o SCR:																									
Value	Explanation																								
0	No aplicable / Desconocido																								
1	Potencia activa																								
2	Potencia Reactiva																								
3	Factor de potencia																								
4	Voltaje																								
5	Precio por la potencia activa																								
6	Precio por la potencia reactiva																								
7	Calor																								
99	Otro																								



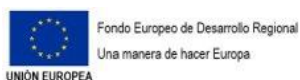


Logical Node (LN)	Data Object (DO)	Explanation	Value
LN CSWI	NamPlt	Placa de identificación	
	Beh	Comportamiento	
	Health	Salud (estado)	
	Mod	Modo	
	LocKey	Local o remoto	
	Loc	Control local del comportamiento	
	OpOpn	Operación "Apertura del interruptor"	
	SelOpn	Selección "Apertura del interruptor"	
	OpCls	Operación "Cierre del interruptor"	
	SelCls	Selección "Cierre del interruptor"	
	OpCntRs	Contador de operaciones reseteable	
	Pos	Interruptor, general	



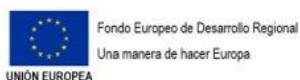


Logical Node (LN)	Data Object (DO)	Explanation	Value
LN DCRP	NamPlt	Placa de identificación	
	Beh	Comportamiento	
	Health	Salud (estado)	
	Mod	Modo	
	LocKey	Local o remoto	
	Loc	Comportamiento de control local	
	OpCnt	Contador de maniobras	
	Pos	Posición del interruptor	
	BlkOpn	Apertura del bloque	
	BlkCls	Cierre del bloque	





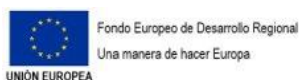
Logical Node (LN)	Data Object (DO)	Explanation	Value
LN DPVM	NamPlt	Placa de identificación	
	Beh	Comportamiento	
	Health	Salud (estado)	
	Mod	Modo	
	AVCrv	Índice en el punto activo de la curva AV	
	MdulCfgTyp	Tipo de configuración del módulo PV:	
		Value	Explanation
		0	Desconocido/ no aplicable
		1	Flate plate
		2	Concentrado
		False	No conectado eléctricamente al ECP
	MdulAVCrv	Curva Amp-Volt del módulo en STC	
	MdulWRtg	Módulo de potencia nominal en vatios pico STC	
	MdulW200Rtg	Módulo de potencia nominal en vatios pico a 200 W/m <sup>2</sup>	
	MaxMdulV	Voltaje del módulo a máxima potencia en STC	
	MaxMdulA	Corriente del módulo a máxima potencia en STC	
	MdulOpnCctV	Voltaje del módulo en circuito abierto	
	MdulSrtCctA	Corriente del módulo en cortocircuito	
	MdulWTmpDrt	Porcentaje de reducción de la potencia en función de la temperatura (potencia/temperatura) por encima de 25°C	
	MdulATmpDrt	Porcentaje de reducción de la corriente en función de la temperatura (corriente/temperatura) por encima de 25°C	





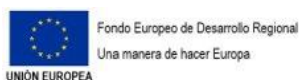
	<b>MdulVTmpDrt</b>	Porcentaje de reducción de la tensión en función de la temperatura (tensión/temperatura) por encima de 25°C	
	<b>MdulAgeDrt</b>	Porcentaje de envejecimiento del módulo con el tiempo	

Logical Node (LN)	Data Object (DO)	Explanation	Value
LN DPVA	<b>NamPlt</b>	Placa de identificación	
	<b>Beh</b>	Comportamiento	
	<b>Health</b>	Salud (estado)	
	<b>Mod</b>	Modo	
	<b>Typ</b>	Tipo de montaje:	
		Value	Explanation
		0	Desconocido/ no aplicable
		1	Matriz
		2	Sub-matriz
		3	Serie
		4	Modulo
		5	Planta
		99	Otro
	<b>ModulCnt</b>	Número de módulos por serie	
	<b>StrgCnt</b>	Número de series en paralelo por sub-matriz	
	<b>SubArrCnt</b>	Número de sub-matrices en paralelo por matriz	
	<b>ArrArea</b>	Área de la matriz	





	GrnConn	Tipo de conexión a tierra:		
		Value	Explanation	
		0	Desconocido/ no aplicable	
		1	Puesta a tierra positiva	
		2	Puesta a tierra negativa	
		3	Sin puesta a tierra	
		99	Otro	
	ArrWRtg	Potencia nominal del array (vatios pico – vatios p)		
Tilt	Montaje de inclinación fija - grados desde la horizontal (puede ser ajustado por estación)			
Azi	Azimut de montaje – grados desde el norte verdadero			

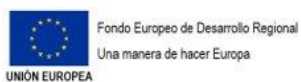




Logical Node (LN)	Data Object (DO)	Explanation	Value	
LN DPVC	NamPlt	Placa de identificación		
	Beh	Comportamiento		
	Health	Salud (estado)		
	Mod	Modo		
	CtrModSt	Estado del modo de control de la matriz		
	ArrModCtr	Modo seleccionado para el control de la potencia de salida de la matriz:		1
		Value	Explanation	
		0	Desconocido/ no aplicable	
1		Seguimiento del punto de máxima potencia (MPPT)		
2		Control del límite de potencia		
3		Limite de la corriente DC		
4		Control del voltaje de la matriz		
99	Otro			

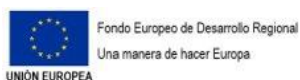


Logical Node (LN)	Data Object (DO)	Explanation	Value
LN MMDC	NamPlt	Placa de identificación	
	Beh	Comportamiento	
	Health	Salud (estado)	
	Mod	Modo	
	Watt	Potencia	
	Amp	Corriente	
	Vol	Voltage	





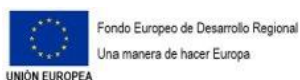
Logical Node (LN)	Data Object (DO)	Explanation	Value
LN ZBAT	NamPlt	Placa de identificación	
	Beh	Comportamiento	
	Health	Salud (estado)	
	Mod	Modo	
	OpTmh	Tiempo de operación	
	TestRsl	Resultado de las pruebas de batería	
	BatHi	Batería alta (voltaje o carga - sobrecarga)	
	BatLo	Batería baja (voltaje o carga)	
	Vol	Voltaje de baterías	
	VolChagRte	Tasa de cambio del voltaje de baterías	
	Amp	Corriente de drenaje de la batería	
	BatTest	Start del test de batería	
	LoBatVal	Valor de alarma de batería baja	
	HiBatVal	Valor de alarma de batería alta	







Logical Node (LN)	Data Object (DO)	Explanation	Value	
LN ZBTC	NamPlt	Placa de identificación		
	Beh	Comportamiento		
	Health	Salud (estado)		
	Mod	Modo		
	ChaV	Voltaje de carga		
	ChaA	Corriente de carga		
	BatChaTyp	Tipo de carga de batería:		
		Value	Explanation	
		0	Desconocido/ no aplicable	
		1	Voltaje constante	
		2	Corriente constante	
		99	Otro	
ChaCrv	Curva de carga			
ReChaRte	Tasa de recarga			
BatChaPwr	Potencia requerida para la carga de la batería			



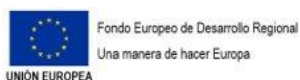


Logical Node (LN)	Data Object (DO)	Explanation	Value
LN ZRCT	NamPlt	Placa de identificación	
	Beh	Comportamiento	
	Health	Salud (estado)	
	Mod	Modo	
	IsoTyp	Tipo de aislamiento:	
		Value	Explanation
		0	Desconocido/ no aplicable
		1	Transformador de frecuencia de potencia aislado
		2	Hi transformador de frecuencia aislado
		3	Sin aislamiento, puesta a tierra
		4	Sin aislamiento, aislamiento de la fuente DC
		99	Otro
	VRegTyp	Tipo de regulación de voltaje:	
		Value	Explanation
		0	Desconocido/ no aplicable
		1	Regulación de salida: voltaje fijo
		2	Regulación de salida: Voltaje variable
		3	Filtrado de salida: Dependiente de la carga
		4	Sin regulación y sin filtrado
		99	Otro



			Tipo de sistema AC:									
	ACTyp		<table><tr><th>Value</th><th>Explanation</th></tr><tr><td>1</td><td>Fase única</td></tr><tr><td>2</td><td>Dos fases</td></tr><tr><td>3</td><td>Tres fases</td></tr></table>	Value	Explanation	1	Fase única	2	Dos fases	3	Tres fases	
Value	Explanation											
1	Fase única											
2	Dos fases											
3	Tres fases											

Logical Node (LN)	Data Object (DO)	Explanation	Value
LN MMET	NamPlt	Placa de identificación	
	Beh	Comportamiento	
	Health	Salud (estado)	
	Mod	Modo	
	EnvTmp	Temperatura ambiente	
	EnvHum	Humedad	
	DffInsol	Irradiancia difusa	
	DctInSol	Irradiancia normal directa	
	HorInsol	Irradiancia total horizontal	
	HorWdDir	Dirección del viento horizontal	
	HorWdSpd	Velocidad del viento horizontal	
	EnvPres	Presión barométrica	







Universidad de Alcalá  
Escuela Politécnica Superior



ESCUELA POLITECNICA  
SUPERIOR



Universidad  
de Alcalá